Thèse n° 9689

EPFL

An Autonomous Robotic Platform to Manipulate Collective Behavior in Ants

Présentée le 27 février 2023

Faculté des sciences et techniques de l'ingénieur Laboratoire de Systèmes MicroBioRobotiques Programme doctoral en robotique, contrôle et systèmes intelligents

pour l'obtention du grade de Docteur ès Sciences

par

Matthias RÜEGG

Acceptée sur proposition du jury

Prof. K. Aminian, président du jury Prof. M. S. Sakar, Prof. L. Keller, directeurs de thèse Prof. Y. Ulrich, rapporteuse Prof. S. Berman, rapporteuse Prof. F. Mondada, rapporteur

 École polytechnique fédérale de Lausanne

2023

Acknowledgements

I thank my co-advisors Prof. Selman Sakar and Prof. Laurent Keller for giving me the opportunity and unlimited freedom to pursue my project. Their steady support was essential for me.

I thank Dr. Alexandre Tuleu, R&D Engineer at the Department of Ecology and Evolution at the University of Lausanne, for his work on the automated *Formicidae* tracking system which is the technical foundation of the robotic manipulation platform developed in this thesis (Chapter 2). His natural emphasis on engineering rigor, documentation and open source mentality is fundamental to his collaborators.

I thank Dr. Alba Motes Rodrigo, Postdoctoral researcher at the Department of Ecology and Evolution at the University of Lausanne, for her collaboration on the experiments on the behavioral response of ants to the robotic nest intruder (Chapter 3). Her contribution made of this chapter an original and stand-alone piece of research.

I thank Prof. Ömür Arslan, Assistant Professor of Robotics at Eindhoven University of Technology, for his collaboration on the socially aware robot navigation algorithm for crowded environments (Chapter 4). In countless meetings, Ömür has been one of my most valuable sources of advice in terms of clarity, methodology and constructive criticism.

I thank Tomas Kay, Thomas Richardson, Nathalie Stroeymeyt, Sean McGregor, Céline Stoffel, Christine La Mendola, Erik Frank, Falon Pasquier and Jason Buser of the Department of Ecology and Evolution at the University of Lausanne for introducing me to their field and for their advice at any time.

I thank Erik Mailand, Ece Özelçi and Fazil Uslu of the Microbiorobotic Systems Laboratory at EPFL for their collaboration on various projects that allowed me to gain valuable experience as an engineering.

I thank everyone who has helped me personally and outside of the strictly professional domain. Those are most of the above mentioned and many more, but above all my partner Antonia.

Lausanne, February 9, 2023

M. R.

Abstract

Eusocial life is characterized by division of labour, collective decision making and self organization, and regarded as the highest form of social organisation in groups. Ants are a model organism for research in collective behavior and the evolution of eusociality. Recently, mobile robots have been developed as an experimental tool to investigate animal behavior in manipulative interactive experiments with mammals, fish, birds and bees. Manipulating the collective behavior of an eusocial organism at the individual level is instrumental to study the mechanisms of self-organization. In this thesis, the Antbot, a robotic manipulation platform, is developed with an ant-sized robotic dummy to deliver a tactile stimulus to a target individual without interacting with any other ant. The system is integrated with a state-of-the-art tracking system to provide real-time visual feedback for teleoperated and autonomous manipulation. We use the Antbot in an experiment to investigate the social factors that influence the levels of individual aggression and responsiveness. Aggression is a central behavior in animals, particularly in eusocial insects, as it serves to maintain the colony integrity by protecting resources such as brood, food, territory and workforce. We found that aggression and responsiveness to tactile stimuli depend on the previous activity of the ants, the social context, and the individual task profile. We also demonstrated a habituation effect in individuals that were previously more active. As a milestone towards automated experimentation, a navigation algorithm for socially aware navigation in dynamic crowded environments is developed and implemented on the platform. A software framework is developed for real-time processing of social parameters, such as interaction network measures, that provide the basis for experimental automation with easily re-programmable behavior for the robot. The autonomous operation of the Antbot platform is tested with a proof-of-concept experiment.

Keywords: Animal behavior, Robotics, Ants, Animal-robot interaction, Robot navigation

Zusammenfassung

Eusoziales Leben ist durch Arbeitsteilung, kollektive Entscheidungsfindung und Selbstorganisation gekennzeichnet und gilt als die höchste Form der sozialen Organisation in Gruppen. Ameisen sind ein Modellorganismus für die Erforschung des kollektiven Verhaltens und der Evolution von Eusozialität. Als experimentelles Werkzeug wurden in jüngster Zeit mobile Roboter entwickelt, um das Verhalten von Tieren wie Säugetiere, Fische, Vögel oder Bienen in manipulativen interaktiven Experimenten zu untersuchen. Die Manipulation des kollektiven Verhaltens eines eusozialen Organismus auf individueller Ebene ist ein wichtiges Instrument zur Untersuchung der Mechanismen der Selbstorganisation. Das Ziel dieser Arbeit ist die Entwicklung eines Experimentalaufbaus, genannt Antbot, mit einem mobilen Roboter als ameisengroße Attrappe, um bei einem Individuum in einer ungestörten Ameisenkolonie einen taktilen Reiz zu erregen. Der Antbot ist integriert in ein state-of-the-art Tracking-System für Echtzeit-Feedback im teleoperierten oder autonomen Betrieb. Es besteht aus einem Robotermanipulator zur Fortbewegung und Signalerzeugung und einer magnetisch geführten Attrappe ("Dummy"), dessen Ausführung je nach Bedarf für alternative Reize modifiziert werden kann.

Zur Untersuchung der sozialen Faktoren, die das Ausmass der individuellen Aggression und Reaktionsfähigkeit beeinflussen, setzen wir den Antbot in einem Experiment ein. Aggression ist insbesondere bei eusozialen Insekten ein zentrales Verhalten, da sie dazu dient, die Integrität der Kolonie aufrechtzuerhalten und so den Verlust von Ressourcen wie Brut, Nahrung, Territorium oder Arbeitsleistung zu vermeiden. Mit dem Antbot haben wir festgestellt, dass Aggression und Reaktionsfähigkeit auf taktile Reize von der vorangehenden Aktivität der Ameisen, dem sozialen Kontext und dem individuellen Aufgabenprofil abhängen. Ferner konnten wir bei Individuen die zuvor aktiver waren einen Gewöhnungseffekt nachweisen.

Zusammenfassung

Als Meilenstein hin zur Autonomie wird zudem ein Algorithmus für die Roboternavigation in dichten dynamischen Umgebungen entwickelt und auf der Hardware implementiert. Es wird ein Software-Framework für die Echtzeitverarbeitung sozialer Parameter, wie z. B. Interaktionsnetzwerke, entwickelt, das die Grundlage für eine experimentelle Automatisierung mit leicht umprogrammierbarem Verhalten des Roboters bildet. Die Autonomie des Antbot wird in einem demonstrativen Experiment getestet.

Schlüsselbegriffe:

Verhaltensbiologie, Robotik, Ameisen, Tier-Roboterinteraktion, Roboternavigation

Contents

Acknowledgements					
Ał	ostra	ct (English/Deutsch)	iii		
In	trod	uction	1		
1	Bac	kground	5		
	1.1	Social behavior in ants	5		
	1.2	Automated experimental techniques	7		
		1.2.1 Tracking individual ants	7		
		1.2.2 Motorized manipulation systems	9		
		1.2.3 Automation and closed-loop manipulation	12		
	1.3	Mobile agents as manipulation tools	13		
2	Rob	ootic manipulation platform	17		
	2.1	FORT tracking system	19		
	2.2	Model experiment and requirements	22		
	2.3	Hardware setup	24		
	2.4	Software	29		
		2.4.1 Processing of social proxies	29		
		2.4.2 FORTloop: experiment automation	32		
		2.4.3 Autonomous interactive experiments	36		
	2.5	Experimental validation	41		
		2.5.1 Towards autonomous manipulation	44		

Contents

	2.6	Conclusion	49				
3 Tactile stimulation of individuals in an ant colony							
	3.1	Introduction	51				
	3.2	Methods	54				
		3.2.1 Robotic manipulation system	54				
		3.2.2 Experimental set-up	54				
		3.2.3 Experimental design	55				
		3.2.4 Data collection and coding	56				
		3.2.5 Statistical analyses	58				
	3.3	Results	61				
		3.3.1 Factors influencing the probability of responding to the dummy	61				
		3.3.2 Factors that influence the probability of alarm behavior	67				
		3.3.3 Habituation hypothesis	69				
	3.4 Discussion						
		3.4.1 Conclusion	74				
4	Socially aware robot navigation						
	4.1	Introduction	76				
	4.2	Simplicial sphere world complexes	80				
		4.2.1 Definitions and notation	81				
		4.2.2 Polygonal sphere world	84				
		4.2.3 Simplicial sphere world	84				
	4.3 Connectivity of regular simplicial sphere worlds						
	4.4	High-level planning in simplicial sphere world complexes	96				
		4.4.1 Motion graph	96				
		4.4.2 Path re-identification	99				
	4.5	Simulations	100				
		4.5.1 Local motion planner	100				
		4.5.2 Evaluation metrics	104				

	4.5.3 Multiple agents with conflicting paths					
		4.5.4	Single-agent in a dynamic environment	107		
		4.5.5	Software	109		
	4.6	Discu	ssion	110		
5	5 Discussion					
	5.1	Techn	ical challenges	113		
	5.2	Outlo	ok	114		

A	Арр	pendix 11						
	A.1	Antbot software architecture and implementation						
		A.1.1	Subprocess state machines	118				
		A.1.2	Performance conisderations	119				
	A.2	Exper	imentation protocols	121				
		A.2.1	Tagging (without anesthetization)	121				
		A.2.2	Robotic dummy fabrication	122				
		A.2.3	Experiment preparation	122				
	A.3	Defin	itions for mesh construction	128				
		A.3.1	Simplex, face and facet	128				
		A.3.2	Piecewise linear complex	128				
		A.3.3	Quarantined complex	129				
	A.4	Mesh	construction algorithm	130				
		A.4.1	Construction of constrained regular triangulations	130				
		A.4.2	Triangulation refinement	134				
		A.4.3	Implementation details	135				
		A.4.4	Computation of the orthosphere	136				
	A.5	Increi	mental build regular triangulation python package	137				

Bibliography

Contents

Curriculum Vitae

Introduction

Motivation: Social life and the engineering perspective

The organization of social life fascinates people of every background, and provides a rich source of inspiration for roboticists. Ant colonies are a prime example of a superorganism, capable of solving complex tasks collectively with limited individual capabilities and information (Hölldobler et al., 1990). The evolutionary advantage of this collective intelligence manifests itself in the ubiquitous abundance of ants on the planet. A recent study has estimated the total number of ants on earth as 20×10^{15} , and the total ant biomass to exceed that of birds and mammals combined (Schultheiss et al., 2022). Through self-organization, individual ants form a superorganism with spatially organized task division (Mersch et al., 2013), robustness in task solving (Feinerman et al., 2018) and resilience towards perturbations and hazards (Stroeymeyt et al., 2018).

The collective behavior of ants has been a source of inspiration for solving complex engineering problems with distributed algorithms running on simple machines. An example on one end of the spectrum of organizational complexity are the leaf-cutter ants. These ants farm fungus with highly evolved task division that is in some species functionally linked to a big variety in worker morphology (Hölldobler and Wilson, 2010). On the other end of the spectrum we see division of labour and task specialization emerging in groups with fewer than 10 individuals with very similar appearance (Ulrich et al., 2018). This is great news to the designer of collaborative robots, as the prerequisites are, in an abstract sense, are quite similar. Bio-

Introduction

inspired engineering designates the effort to design applications by adopting some of these strategies from biological observations. Examples range from collective transport of physical robots (Berman et al., 2011) to algorithms for non-convex optimization (Dorigo et al., 2006), communication routing (Di Caro and Dorigo, 1998) or robot task allocation (Krieger et al., 2000). While acknowledging the great potential of learning from biological systems, this thesis is primarily dedicated to building a robotic tool to help researchers understand the mechanisms of social life rather than to mimic it.

Problem formulation: Precise manipulative experiments

Studying self-organization of social organisms in their native environment is extremely challenging (Mitri et al., 2013). Complex phenomena may emerge from the interplay of inert sub-units, yet self-organization in social organisms is inherently more complex since the individuals have their own behavioral state (Camazine et al., 2001a). Traditionally, advancements in this field have relied on carefully designed experiments that are done under controlled laboratory conditions (Mitri et al., 2013). Preserving the essential context of the natural environment is however crucial to investigate the authentic social behavior of the colony. To reach a consolidated conclusion, observations must be carried out for long duration and under realistic conditions. In experimental research, manipulation is the only way to demonstrate causality. Robotic technology brings precision, dexterity, and repeatability to manipulation, and may eliminate experimental bias. The overall goal of this thesis is to develop a robotic manipulation tool that can be used to perform experiments with behaving ants in their social context. The thesis aims to create the basis for experimental automation using interactive mobile robotic agents in behavioral experiments with eusocial insects.

Aggression is a fundamental behavior of social organisms. For the survival of the colony, ants must direct their aggression towards intruders. Only this way, they can protect their offspring and livestock. It is also a simple behavior to characterize and detect. It is well-known that pheromones and olfactory signals play a central role but the mechanisms that control aggression in ants is not fully understood (Akino, 2008; Sturgis and Gordon, 2012; Guerrieri et al., 2009). We ask the following fundamental questions to show the capabilities of our robotic manipulation system: What is the repertoire of behavioral responses of individual ants in a colony towards an interactive mobile robotic agent? And how does social context influence these individual responses?

Methodology and major contributions

The *Antbot* is developed as an experimental tool for individual level manipulations of ants within the undisturbed social environment. The system is built upon the infrastructure and capabilities of our automated ant tracking system. As a design objective, we desired the manipulation to be versatile with respect to the type of stimulus. The solution consists of a mobile electromagnetic control module that navigates an ant-sized magnetic agent to desired locations inside the colony where the stimulus is delivered. The current version of the agent is capable of applying a tactile antennation-like stimulus. The agent can be exchanged on the fly during experiments.

With aggression as a first noncommittal area of interest, we present original experiments provide novel insight into the research question stated above, and therefore validates the concept of the Antbot as a useful tool for fundamental research in ant behavior. The thesis presents milestones towards running fully autonomous behavioral experiments, where manipulation is carried out based on a programmed hypothesis considering social factors that are inferred in real time. A software framework processes tracking data in real time, generates the robot behavior that implements the experimental plan, and navigates the agent autonomously. Path planning in crowded dynamic environments is a state of the art problem in robotics research, which must be properly addressed for the successful implementation of automated micromanipulation. A novel algorithm is presented that can achieve persistent and socially compliant navigation for the magnetic agent. This algorithm is then implemented to command the control module of the Antbot. The capabilities of the integrated autonomous

Introduction

Antbot is demonstrated with case studies.

Thesis organization

Chapter 1 gives a short introduction of the social life in ants and summarizes common techniques that were successfully used in manipulative experiments. Experimental automation is motivated and the concept, state of the art and challenges of mixed robot-animal societies is introduced. In chapter 2 the Antbot robotic manipulation platform for autonomous experiments with ant colonies is presented. Chapter 3 reports an experiment with the model species *Leptothorax acervorum*, in which individuals were targeted with a tactile stimulus by a teleoperated robotic dummy and their behavioral response was analyzed with respect to the social context. Chapter 4 introduces a novel algorithm for navigation in crowded dynamic environments like ant colonies, as a milestone towards fully autonomous experiments. Finally, Chapter 5 outlines future biological questions to be addressed with the present system and technical challenges.

1 Background

1.1 Social behavior in ants

Ants live in large groups, called colonies, that often consist of a single reproductive individual, the queen, and a bigger number of sterile workers. In addition to this reproductive division of labour, colony members cooperatively take care of their offspring, the brood, forage for food, or defend the nest. This highly evolved form of social life is called *eusociality*. Although the reproductive individuals are called queen, it is widely accepted that division of labour and eusociality emerge through self-organization, as a result of individual actions and without a central governor (Keller and Chapuisat, 2001). In this respect, ant colonies, along with bees, termites and some other animals, are organizationally distinct from fish swarms or bird flocks. A great challenge in research on eusocial animals is that their behavior can only be studied when the social structure is intact, observations of on an isolated ant have limited value.

Aggression is one of the basic social behaviors and serves to maintain the safety and integrity of the colony, and, not surprisingly, is ubiquitous in the ant world.¹ It manifests not only between individuals of different colonies, but also among nest mates. From an evolutionary perspective, conflict and aggression play a central role in the self-organization of eusocial insects colonies. The context repertoire of aggressive responses ranges from defensive attacks

¹Popular science book on the topic: "Les guerres secrètes des fourmis", Cleo Bertelsmeier, Edition Favre, 2019

Chapter 1. Background

against intruders such as predators, parasites and ants that belong to another colonies, to the regulation of the right to reproduce by mutilation of the reproductive organs of other fertile individuals in certain species of *Diacamma* (Annagiri, 2021). A more detailed account of the role of aggression in ants will be given in Chapter 3, together with a manipulative experiment that centers around aggressive behavior.

How ants recognize nest mates vs intruders and modulate aggressive behavior is a question closely linked to our understanding of signaling, perception and communication in ant colonies. It is known that chemical signals, called *pheromones*, are the primary carriers of information. Pheromones can be deposited or sprayed purposely to mark a location or an event. They are also used passively on the ant 'skin', the *cuticle*, as a means of displaying colony identity. Using their antennae, ants can detect and identify these chemical compounds with extreme sensitivity and pinpoint accuracy. The visual capabilities are very limited in most species; ants spend in fact a large portion of their life in complete darkness inside their nest. In addition to chemical signals, tactile signals are believed to facilitate communication for example when food is exchanged or in tandem running, a behavior which is used by colony members to lead others to a new nest site or food source (Hölldobler et al., 1990).

Manipulative experiments using artificial stimuli posses the power to demonstrate causality, but are intrusive by nature. Studies targeted at understanding the mechanisms and context of individual aggressive behavior have used manipulative techniques, in which alien individuals were mixed with sub-colonies to register a change in behavior with respect to prior acclimatization (Guerrieri et al., 2009) or setups with restrained individuals that were presented with chemical stimuli to measure the *mandible opening response*, a proxy for aggression (Guerrieri and d'Ettorre, 2008). Despite the mechanistic insights that such tests offer, these studies are performed with individual ants that were not in their undisturbed social context. The extent to which individuals are embedded in an environment that they can sense and modify is described by the term *situatedness* (Mitri et al., 2013). Compared to solitary insects like fruit flies, studying eusocial insects poses an exceptionally high standard for manipulative studies with respect to situatedness. A minimally invasive and precisely targeted method is

therefore required to deliver a stimulus, for example to trigger aggression, to an individual in a colony while preserving its social organization during the experiment. A more realistic stimulus is expected to increase the situatedness in the experiment, which means that less experimental factors are controlled. As a consequence, the behavioral response is expected to be more subtle and more complex to analyze. This requires more repetition and longer observation periods to allow statistically significant conclusions. A particular difficulty in using artificial stimuli is to understand its effect on the animal in order to understand its response (Lahti, 2015). It is therefore advisable to start with stimulus as simple as possible, but in an environment as realist as possible, before moving on to attempt a biomimetic approach, in order to understand whether the stimulus is for examples a replacement of an existing stimulus or whether it is completely new.

1.2 Automated experimental techniques

Automating protocols for recording and manipulating ant behavior have a number of advantages, including significantly increased throughput, reduced error and bias, and repeatability. *Tracking* refers to the inference of the positions and, preferably, the pose of individuals, where pose refers to any additional geometric information on the body posture of the individuals on top of the location and orientation of a representative identification point on the body (Dell et al., 2014). For technical and practical reasons, tracking is usually performed on ants that are studied in the laboratory.

1.2.1 Tracking individual ants

Ants have been cultivated under laboratory conditions for a long time. Notably, there are century old protocols in the literature (Forel, 1923). Ants, like most other insects, have a passive respiratory system, which makes them sensitive to humidity and environmental conditions general. To recapitulate their natural living conditions, ants must be kept under controlled ambient humidity, illumination, and temperature. The first aspect of automation is maintaining

Chapter 1. Background

the natural humidity and temperature at all times. To this end, the observation chamber can be placed inside a automated incubation system that is programmed to vary illumination and ambient humidity and temperature, allowing to simulate a daily cycle (Mersch et al., 2013).

The second aspect of automation is the continuous recording of individual ant movements and interactions. Time-lapse imaging and visual tracking of individual ants is instrumental to study their collective behavior. There are three techniques to identify ants: paint, fiducial identification labels or marker-less techniques based on visual recognition. Color coded labelling has been used long before automated tracking became available. For studies that require very long-term identification, e.g. for age marking over the entire lifespan (Kay et al., 2022), color marking would be preferable as it is more durable than tagging with fiducial markers. For colonies with relatively small size, color marking can be used for automated tracking. Color marking is less restrictive for ants with very small body size (Gal et al., 2020). Tracking methods using fiducial markers that are glued on to the ants became available approximately a decade ago (Mersch et al., 2013). The use of identification tags is more laborious to prepare and more restrictive for the ants, but has the advantage of ambiguity free identification over time, which is crucial to build social abstractions including interaction networks. For practical reasons imposed by the tagging procedures with fiducial markers, ant colonies are limited to roughly 500 individuals. Species that form colonies with with a higher minimal individual count are not suited for tracking experiments. Marker-less tracking is to date mainly used in scenarios with a focus on detecting events with limited temporal significance and in simpler settings with fewer individuals. While marker-less tracking requires less experimental preparation and no intervention on the organism, the accompanying computational techniques are intensive, making real-time tracking and identification very challenging. In all cases, imaging relies on optical access to the nest. For good detection, the species must be able to live essentially on the surface and inhabit preexisting structures to avoid visual obstruction. Therefore, ants that need to dig their nests inside soil are excluded from such studies.

Different ant species may have a very different lifestyle. In addition to the colony architecture and size, the overall activity patterns, social structure and many other characteristics can greatly vary (Parr et al., 2017). The robotic micromanipulation platform presented in this thesis is built on top of the *Formicidae Tracking System (FORT)*, a state-of-the-art automated video tracking system developed in our group at UNIL. FORT is described in more detail in Section 2.1. Table 1.1 contains a list of species that have been successfully tracked with this system to give an idea about species characteristics that determine the complexity of the environment to operate a mobile agent. The values were computed based on a day of tracking data of a colony. Some colonies occupy a larger space in the system with one or even two coupled arenas for foraging and nest (L or L-L configuration in Table 1.1). Other species prefer a covered nest inside a single arena, which is linked to their natural nesting behavior, where they live under the bark of dead trees (*L. acervorum*) or under stones (*L. niger*). As a consequence, the nesting behavior of *L. niger, L. acervorum, T. nylanderi* and others requires a covered nest with low ceiling (1-2mm). The geographical distribution of a species can be of practical consideration, as it is easier to collect endemic species, which is instrumental to perform quick tests during the development of a new manipulation technique such as ours.

Tracking can be performed online or offline on recorded images. Real-time processing provides higher standards for tracking accuracy and resolution. However, processing images in real-time bears a heavy computational load. With real-time reactive manipulative systems that recently have gained popularity in behavioral research, online tracking has obtained an additional practical value. Data becomes available for further processing while the experiment is running, providing a basis for automated manipulative interventions. Recent work has introduced an automated framework where a reward is presented upon detection of a particular action to condition behavior (Kane et al., 2020).

1.2.2 Motorized manipulation systems

Manipulative experiments on animal behavior are often performed using artificial stimuli (Tinbergen, 1951). The organisms that have been studied this way range from birds to ants, which demonstrates the benefit of manipulation as a strong test for causality. A stimulus can be direct, such as the presentation of dummy egg to a bird, or indirect like the change of an

species	subfamily	colony size	worker size [mm]	speed [mm/sec]	$\Delta \theta / \Delta t \ [^{\circ} / \text{sec}]$	occupancy [%]	time active [%]	arena configuration	data source
Leptothorax acervorum*	Myrmicinae	160	3.7	1.1	35.7	47.1	7.5	S(N)	this thesis
Temnothorax nylanderi*†	"	136	2.85	0.4	23.1	30.0	10.6	S(N)	Richardson et al. (2022)
Pogonomyrmex barbatus	"	104	8.5	5.3	63.0	27.5	42.9	L-L	Kay et al. (2022)
Lasius niger*†	Formicinae	102	4.25	1.1	38.9	25.8	14.9	S(N)	Richardson et al. (2022)
Camponotus fellah	"	96	6.5	1.6	33.4	60.0	60.9	L-L	Kay et al. (2022)
Camponotus floridanus	"	68	7.8	8.0	78.2	25.1	34.1	L(N)	Kay et al. (2022)
Diacamma rugossum	Ponerinae	95	10.3	3.9	49.5	40.2	29.5	L-L	Kay et al. (2022)
Iridomyrmex purpureus	Dolichoderinae	95	7.9	5.0	29.5	70.5	37.2	L-L	Kay et al. (2022)
Rhytidoponera metallica	Ectatomminae	104	6.1	2.8	61.5	25.0	26.8	L-L	Kay et al. (2022)
Apis mellifera*	Apidae/Apis	81	13.4	11.1	106.1	59.8	60.8	S(N)	Liberti et al. (2022)

Table 1.1: Tracking characteristics of ant species tracked with FORT and (†) predecessor system (Mersch et al., 2013). Colony size reported for specific experiment, can vary substantially in the wild. Time active: when moving faster than 5% of body size per second. $\Delta\theta/\Delta t$: directional change, angle between two segments of ant trajectory. Speed and directional change are only measured during active period. Time active, speed and directional change are colony averages. Occupancy: Average percentage of nest space occupied by ants, area of union of disks representing ants divided by convex hull area of all ants in nest. Arena configuration: S: Small arena [105 × 145] mm, (N) integrated nest space [65 × 95] mm or 90 mm \emptyset for *C. floridanus*, L: Large arena [170 × 220] mm, L-L: nest/foraging arena double box setup. Values calculated from 24 h tracking data. (*): endemic in Switzerland.

environmental parameter. A famous example in ants is the study of trail formation to exploit different food sources. The question spawned a variety of manipulative experiments, whereby paths to different food sources were offered under laboratory conditions and manipulated to observe the adaptation of the collective foraging behavior. Such experiments allowed to build testable models, which describe the collective foraging success based on the environmental factors and a small number of colony specific behavioral parameters (Camazine et al., 2001b).

Studying social behavior is very challenging due to the manifold of known and unknown factors. In some situations, reducing the dimensionality of the experiment or the number of choices of the organism by restricting its movement is a valid strategy to cope with this complexity. This has been done for example in the study of collaborative transport in ants, where the food item to be transported was attached to a shaft or string, in order to reduce the directional decision to left or right (Feinerman et al., 2018). In an experiment with fish schools in interaction with a robot dummy, the aquarium was built in the shape of a ring to restrict the

swimming direction to clock- or counterclockwise, making it simpler for the robotic agent to provoke a predictable response (Bonnet et al., 2018) (Figure 1.2, center).



Figure 1.1: Motorized setups in ant behavioral research. Left: rotating nest geometry (Heyman et al., 2019), Center: moving nest wall to extend physical distance between two chambers (T. Richardson et al., unpublished). Right: floor heating patches to provoke evacuation behavior (Gal and Kronauer, 2022)

Manipulative systems with the possibility to change the stimulus automatically during the experiment have been developed for many species including ants. A modern version of an experiment to understand the orientation capabilities of ants with respect to food sources and the nest was developed in (Heyman et al., 2019). A rotating nest is used to decouple the effect of visual and chemical cues in ant navigation (Figure 1.1, left). In an unpublished study of the Keller group at UNIL, we have used a partitioned nest to force the colony to segregate into two chambers. This is a common situation in nature for ant species who do not build their nests but inhabit pre-existing structures such as rock crevasses or the bark of a dead trees. Several papers have suggested that social insect colonies are spatially divided into partially overlapping worker groups, and that some of these groups function as *intermediaries*, i.e. as a sort of 'social glue' between the groups. These individuals move back and forth between nurse and forager communities, and in doing so, supposedly maintain the cohesion of the colony. To investigate these intermediaries, we performed a manipulative experiment in which ant colonies were housed in nests that were divided with an automated moving wall to physical manipulate the distance between the groups and test if the switching frequency would be maintained by the intermediaries (Figure 1.1, center).

In (Gal and Kronauer, 2022) a floor heating system was designed to provoke evacuation behavior of the colony as a response to unfavorable temperature conditions (Figure 1.1, right). The fine control of the stimulus allowed to show that collective response is characterized by an emergent sensory response threshold as opposed to an individual one. The calibrated and spatiotemporally controlled application of a stimulus allows to abstract and and test a hypothesis formulated by mathematical models. In (Rajendran et al., 2022), a motorized selective door setup was used to impose a conflict of interest with respect to prospective new nests upon a migrating colony by blocking the access to certain areas for given individuals. In this example, automation was necessary to reduce erroneous manipulations and increase the throughput, as it is infeasible for an experimenter to identify individuals and operate the door simultaneously. Direct artificial stimuli are often designed to target a single perceptive channel, like a sound (De Rosa et al., 2022), a visual illusion generated with a hologram (Stowers et al., 2017), a bio-mimetic dummy (Landgraf et al., 2010) or a local heat source like it was used to attract bees to defined sites inside the hive (Griparić et al., 2017). In a system designed for Drosophila, a moving dummy was used to provoke an optic flow stimulus for the animal to understand how it reacts to approaching obstacles (Agrawal et al., 2014).

1.2.3 Automation and closed-loop manipulation

Besides increasing the experimental throughput, automation standardizes the application of artificial stimuli. Moreover, automated systems can be programmed to react to a complex input based on which a decision must be taken while respecting experimental constraints (Krause et al., 2011). As a semi-automated manipulation scheme, parts of the process could be automated, while the decision on when, where and how to interact is still left to the experimenter. This can be achieved by using pre-processed data, i.e., the identity of an individual and its current location, as an assistive measure for the operator to decide on the specifics of manipulation. In complex settings, it may become necessary to automate the process of the manipulation itself. This can be especially useful if the manipulation is difficult or the intensity of the stimuli must be controlled with precision. In neurological studies, behavioral patterns are detected from the real-time representation of the animal pose. Manipulation as a reaction to such events requires a low latency. An integrated framework, called *DLClive*, that estimates the pose of the animal in real-time to automatically generate an artificial stimulus has been recently presented (Kane et al., 2020). This framework is very analogous to the *FORTloop* package that is introduced in Chapter 2.

1.3 Mobile agents as manipulation tools



Figure 1.2: Autonomous agents for animal-robot interaction. Left: pheromone coated cockroach robot (Halloy et al., 2007). Center: fish robot with biomimetic morphology (Bonnet et al., 2018, 2017). Right: rat robot with biomimetic posture dynamics and stuffed rat (Shi et al., 2013).

Mobile agents can penetrate deeper into a colony or group of animals, and directly target an individual (Mondada et al., 2013). In (Halloy et al., 2007), a pheromone marked mobile robot was used to target the light avoiding behavior of cockroaches by preoccupying shaded areas, and thereby manipulate its decision making process (Figure 1.2, left). In this example, it was shown that, aside from the robot movement, the biomimetic signal on the chemoreceptive channel is sufficient to provoke a typical behavior. Thus, it is not necessary to mimic the appearance of the cockroach. In (Shi et al., 2013), an autonomous mouse agent is used to investigate the factors for a hostile or friendly response to a perceived behavioral pattern (Figure 1.2, right). In this example, actuation is needed for two reasons. First, the targeted perception is not only a static visual cue, but the recognition of a an articulated posture. To this end, the robot attempts to mimic a behavior, which requires a complex design of the agent itself, include a number of articulated joints. Second, the distance between the behaving animal and robotic agent must be carefully controlled because this distance is expected to act

Chapter 1. Background

as an influencing factor. The bee robot introduced in (Landgraf et al., 2010) imitates the waggle dance. To date, this robot is the only mobile system that has been integrated in a eusocial community. Figure 1.2, center, shows an autonomous fish robot that is mimetic in morphology and movement. It delivers a stimulus to a local range of individuals in a swarm to provoke a collective reaction. This example highlights the importance of tracking and feedback, as the propagation of the signal from one fish to the next by changing its direction of movement is very fast and the robot must be able to follow the swarm to precisely apply its stimulus. In ants, spatial organization plays an important role in the organization of social life (Mersch et al., 2013; Richardson et al., 2022), therefore restrictive geometries to facilitate the access to deliver a stimulus, as described in Section 1.2.2 has limited used compared to a mobile agent. To study the individual response to stimulation in a social context, it is mandatory to preserve the social organization of the colony and to deliver a stimulus in a precise manner to an individual at the location of its own choice.

Navigation of mobile agents in dynamic environments

The colony should be left intact and minimally disturbed, therefore, the stimulus needs to be delivered to a moving target, i.e. an ant, inside an colony while reducing the number of unintended and unnecessary interactions. Although positional data is available on a global level at a given time through the tracking system, little can be said about the motion of the ants in the near future or their reaction to a moving object nearby. Human crowds permit certain simplifications and predictive strategies, e.g. the stream of people who walk in approximately the same direction on a sidewalk. However, ants in their nest often appear to show bursty, random-walk like movement patterns, which are hard to predict. Furthermore, ants communicate mostly by means of chemical cues (as opposed to humans in crowds who visually process their surroundings), especially inside the darkness of the nest. Fortunately, the instantaneous information in the tracking system is almost complete, in the sense that the position and orientation of every individual is known with the exception of short term detection miss due to visual obstruction or reflections.

Teleoperation refers to the process of remotely navigating the mobile agent by a human operator. The ant system with the incubation chamber restricts direct visual and physical access, yet provides a live video stream as feedback for the operator. The behavioral data shown in Chapter 3 is collected by teleoperating the robotic agent. Inaccuracies and variance in the way stimuli are applied by a human operator can cause blurred results or completely prevent experiments if the task is too delicate. If the stimulus has a complex form, like an entire behavioral pattern, as for the example of the rat robot given above, human bias may become a problem. Experimenter bias is the situation when researchers' expectations influence the outcome of the study and can lead to increased effect observation (Holman et al., 2015). During long experiments and many replications, the experimenter may undergo a learning curve, even when the experimenter is unaware of the tested hypothesis (Rosenthal and Rosnow, 2009). As a further motivation for automation, even if the task is as simple as contacting a target individual, teleoperation may take significant time if the path is obstructed or the ants are moving relatively fast. The more complex the task and the environment are, the higher is the need for automation. While technically challenging, closing the loop for automation removes bias, increases throughput and precision, and makes the effects of the applied stimuli more traceable. To the best of our knowledge, no mobile autonomous system for a mixed eusocial robot-animal society exists. In this thesis, the first steps are taken to build such a system for ant colonies.

An important aspect of automated navigation is the development of proper algorithms for motion planning and control. A global path planning algorithm designed for a static world could be implemented. However, the disregard of the dynamic aspects of ant motion may lead to repeated re-planning due to frequent change of ant positions, which may manifest itself as oscillatory agent motion without notable progress towards the goal. In this thesis, we considered motion planning based on the following intuition: a plan should not be changed if the latest optimal solution offers only a slight improvement over the current solution while requiring a drastic correction. The objective is to be as socially-acceptable as possible by minimizing unintended interactions with the ants, avoiding situations where we come too

Chapter 1. Background

close to a non-targeted ant or stay around her vicinity too long, while completing the tasks in a reasonable amount of time. Socially aware navigation will be discussed in more detail in Chapter 4. We formally implemented this intuition in a novel algorithm to achieve a form of temporal persistence in path planning.

2 Robotic manipulation platform

This chapter introduces the hardware and software architecture of the *Antbot*, the robotic manipulation platform that is designed to navigate an ant-sized magnetic dummy inside the colony and deliver a targeted stimulus to individual ants. The Antbot can be manually controlled by an operator using teleoperation or programmed to run an experimental plan autonomously. The real-time tracking data that is used to assist the operator during teleoperation or as a visual feedback for autonomous operation is provided by *Formicidae Tracking System* (FORT). The capabilities and the software interface of FORT is explained in Section 2.1. Section 2.2 lists the specifications of a representative manipulation experiment to motivate the design choices for the remaining of the chapter. The hardware components of the Antbot are described in Section 2.3. Section 2.4 introduces the brain of then Antbot, *FORTloop*, a versatile software framework to standardize and facilitate the design of automated manipulative experiments with FORT.

The trajectory data acquired by FORT can be post-processed to detect social events or infer individual or group social characteristics. For example, social communities can be inferred from the interaction network, which can serve as an online feedback for the Antbot. Section 2.4.1 summarizes these metrics and the code base developed to compute them for post-processing or in real-time.

Automation has the following advantages that are considered during the design of the Antbot:

- 1. Increased precision and repeatability for manipulation
- 2. Increased throughput and longer operation
- 3. Adaptive changes in the experimental plan using real-time tracking data
- 4. Avoidance of human bias that is introduced through learning or cognitive processes

In this thesis, we use a tactile stimulus in the form of an oscillatory movement. The stimulus is exerted by the mobile robotic Antbot dummy through a pair of synthetic antennae that is mounted on its body. The intensity of this signal must be precisely controlled to interpret the behavioral response of the ants depending on the biologically relevant factors. Moreover, stimuli must be delivered to different individuals at their current location in the nest while avoiding interactions with other ants. Navigating the robotic dummy via teleoperation is very challenging due to the complex environment of the colony (e.g., crowded space, ants are moving around) and due to the lag in the visual feedback. Thus, automation can make an important contribution.

Some of the social representations of the colony include the detection of communities, i.e. partitions, in the interaction network or in the site-visiting network. Both data structures are proxies for division of labour. Depending on the actual hypothesis, providing the experimenter with such information is instrumental to include social parameters as factors in the adaptation of the manipulation tasks. Automated data processing is therefore required to enable more complex action planning.

Even when a manipulative experiment is carried out as a blind study, where the experimenter is unaware of the hypothesis tested, a learning curve can be developed in the way the manipulations are performed. Therefore the manipulation is not constant over time, which can bias the behavioral response. This problem and other potential artifacts that introduced by a human experimenter are known issues in behavioral research (Rosenthal and Rosnow, 2009). These effects are more pronounced in experiments involving social interactions with humans or primates, but can also be a problem in experiments with ants. Although it can be desirable that a robotic system learns as well during an experiment (Landgraf et al., 2021), artificial learning is reproducible and can be reenacted and interpreted.

Automation in complex scenarios is challenging and requires a high degree of autonomy. Closing the loop for a fully autonomous experiment might seem to result in a small benefit compared to the programmatic effort. Yet, it promises a further increase of throughput, as experiments could be run around the clock for several days. Automation can also help to remove sources of human error, already in semi-supervised operation, and further offload decisive tasks from the operator by deliberate experimental planning. This chapter presents the efforts taken towards full autonomy.

2.1 FORT tracking system

The FORT system was developed between 2018 and 2020 by Dr. Alexandre Tuleu in the group of Prof. Laurent Keller at University of Lausanne, as an upgrade to the tracking system presented in (Mersch et al., 2013).

The system uses fiducial markers to identify individuals with ultimate accuracy during longterm studies. To date, the longest experiment involved 100 days of consecutive tracking per colony replicate, which lasted the typical lifespan of a worker of the species under investigation (Richardson et al., 2021). Such experiments are needed to investigate transitions in social behavior due to individual maturation. The tracked positions per image, called tracking frame, are saved continuously in a data file and only a compressed version of the video stream is stored for visual inspection. This reduces data to storage, and decreases the risk and amount of data loss due to technical interruptions of the tracking. These advantages were already identified in the first design and proved practical in a number of experiments (Mersch et al., 2013; Richardson et al., 2021; Stroeymeyt et al., 2018).

FORT, the second generation tracking system, inherits the real-time tracking capability from

its predecessor. As an upgrade, it provides easy access to the tracking data through real-time data transfer with network sockets over the local area network (LAN). This new feature opens the doors for closed-loop or human-in-the-loop manipulative experiments, during which the tracking data is further processed online to infer social characteristics of the colony or the individual. A collection of the characteristics used so far is given in Section 2.4.1.

Compared to its predecessor, FORT received additional major upgrades that are essential for the development of the *FORTloop*. Most importantly, the frame rate was increased from 2 Hz, hard coded, to a maximum 26 Hz when used with secondary tracking computers that can be daisy-chained to process a fraction of the visual input in parallel. As an important remark, a frame rate of 8 Hz is sufficient for safe navigation of an autonomous robotic dummy inside the colony and represents the maximal tracking frequency of a single host. The design is more modular and compact, which results in more room for the necessary hardware for the Antbot. Further improvements are the resolution from 12 MPx to 48 MPx and the decoupling of its climate and tracking subsystems, which results in a more fail-safe operation. Notably, the FORT hardware and software is fully opensource.¹

FORT consists of an experimental component with hardware and software for tracking and climate control and a component for essential post-processing of the raw data (Figure 2.1). The tracking PCs, called hosts, store the tracking frames in a serialized data file. *hermes* is the software component to serialize and store (*.hermes* file) or exchange the data among hosts in case the computational load of one experiment is distributed to increase the tracking frequency. The data stream is made available to external clients for further processing (see *py-hermes* component in Figure 2.1 and Section 2.4.2). The post-processing component consists of the *myrmidon* library for data interpretation (interaction detection, trajectory extraction, etc.) and the *fort-studio* graphical user interface for the necessary post processing step of marker alignment. The latter is needed for correct interaction detection. *myrmidon* stores the generated metadata in a file with the same name.

¹https://github.com/formicidae-tracker



Figure 2.1: FORT conceptual overview. *fort-studio*: basic post-processing GUI e.g. for tag alignment and basic data viewer. *myrmidon*: post processing library: interaction detection, trajectory extraction, etc. *hermes*: data exchange protocol. The tracking hardware setup (without host) is shown in Figure 2.3.

The FORT hardware setup along with the Antbot is shown in Figure 2.3. FORT has the capability to regulate the humidity and temperature and the visible light, allowing to simulate a diurnal cycle. The infrared flashlight, which is used to capture images for tracking around the clock, is decoupled from the climate system. The field of vision, given by the camera, is adjusted according to the tag size used.² A smaller tag requires a higher spatial resolution, which in turn reduces the field of vision. The ant arena, the space where the ants live, is a box in the center of the system (blue box in Figure 2.3). This box constraints the motion of the ants to a plane for continuous tracking. The insulating box, needed for climate control, puts further constraints on the workspace regarding the robot integration. FORT makes it technically possible to connect one or more additional arenas via tubes, if necessary.

²A spreadsheet to calculate optical settings for an experiment is found at

 $https://docs.google.com/spreadsheets/d/108 jTYh_x0X60 FvLMI6AnxPJqFTBPCzauivXxzxy9Sjo/edit#gid=0.ptfspc2auivXxy9Sjo/edit#gid=0.ptfspc2auivXxxy9Sjo/edit#gid=0.ptfspc2auivXxxy9Sjo/edit#gid=0.ptfspc2auivXxxy9Sjo/edit#gid=0.ptfspc2auivXxxy9Sjo/edit#gid=0.ptfspc2auivXxxy9Sjo/edit#gid=0.ptfspc2auivXxxy9Sjo/edit#gid=0.ptfspc2auivXxxy9Sjo/edit#gid=0.ptfspc2auivXxxy9Sjo/edit#gid=0.ptfspc2auivXxxy9Sjo/edit#gid=0.ptfspc2auivXxxy9Sjo/edit#gid=0.ptfspc2auivXxxy9Sjo/edit#gid=0.ptfspc2auivXxxy9Sjo/edit#gid=0.ptfspc2auivXxx9Sjo/edit#gid=0.ptfspc2auiv$

2.2 Model experiment and requirements

As a basic model experiment, we chose to investigate the individual factors for showing aggressive individual behavior. Aggression is a basic behavioral response with a central role in social organisms and it is relatively simple to detect. For this purpose, we chose *L. acervorum* with low implications on actuation. The members of this species are relative calm in their activity patterns (Table 1.1) and show low aggression in general (Forel, 1874). Notably, they show a number of intriguing and not well understood behavioral responses (Richardson et al., 2017; Bourke, 1991; Franks et al., 1990). The following specifications are identified:

- R1 (Mobility and workspace) (1) The end-effector must be capable of reaching individuals in a cluttered environment to deliver the stimulus while minimizing unintended interactions. (2) Here we consider manipulations carried out inside the nest, which displays a lower limit on the robotic dummy workspace.
- R2 (Size) (1) The end-effector, i.e. the robotic dummy, must operate inside an ant colony and deliver a tactile stimulus at the approximate ant body height. (2) The manipulator must be integrated in the incubator which displays a limit on all dimensions.
- R3 (Accessibility) In the lab, *L. acervorum* are kept in glass covered nests of 2 mm height under which the robot must operate. The nest and the use of the overhead camera only allow an manipulator operating under the floor.
- R4 (Subtlety) The emanating signal or cues must be subtle enough to measure a gradual behavioral response of the individual. A signal that is too weak or too strong might elicit no response or potentially unnatural behavior in all interactions, respectively.
- R5 (Smell) Ants mostly communicate through chemical signals. It has been shown that the colony odor can be passively acquired by items that are left inside the nest for acclimatisation. To be able to compare acclimated dummies to non-acclimated ones, and for potential future studies with pre-treated dummies, it is important to have the capability to exchange dummies during the experiment. The nest entrance is often

occupied by many individuals. To introduce new dummies without attracting the attention of ants, it is useful to build a nest with a secondary entrance that is only opened during the entry and exit of the robotic dummy. The dummies should not provide any chemical cue and must be treatable to make it chemically inert.

- R6 (Variable stimulus) A priori, the form of the tactile signal is not known. It is therefore desirable to decouple the angular actuation needed for guidance with the magnetic signal applied for the tactile stimulus.
- R7 (Versatile end-effector design) Future studies may use different sized or shaped dummies. Designs that are used to displace items instead of performing interactions might be considered as well.
- R8 (Long term operation) Ants are able to carry heavy payloads and in *Leptothorax* in particular, it is known that colonies improve the nest geometry with lose material found inside the preexisting nest they inhabit (Camazine et al., 2001a). It is therefore desirable that the robotic dummy remains stationary unless it is moved by the system. Long term operation is also a requirement on the software side.
- R9 (Automation and Autonomy) Processing the input and coordinating hardware, either as a visual aid for the operator or to be used as closed loop feedback, requires a software platform capable of (a) Managing the acquisition, pre-processing (tag-heading alignment, interaction detection) and re-distribution of FORT data to (b) Asynchronous sub-processes for target selection/experimental objective planning based on (c) realtime data processing, trajectory planning, hardware communication and data visualization (d) reactive robot control which is fast and easy to reprogram for alternative experimental scenarios.
- R10 (Safe navigation) Certain (polygonal) areas inside the nest should be avoided (brood).Ants that are not currently targeted for a contact with the antbot, should also be avoided.For this purpose we define a safety region, i.e. a large enough disk centered at each ant, with which the robot should minimize collisions. This task can be carried out in by the

operator in manual mode. Automating it represents the challenge of *socially compliant navigation*, which is addressed with the algorithm developed in Chapter 4.

2.3 Hardware setup

Table 2.1 lists the complete hardware for the Antbot. The system consists of three major components: a robotically-actuated mobile *robotic dummy* as the end-effector, an *electromagnetic control system* to navigate the dummy and apply the tactile signal, and the stacked *actuator* on top of which the electromagnetic control system is mounted (2.2).

The dummy moves by sliding on the Teflon[™]covered surface of the nest arena. In this basic version, it is intended to deliver tactile stimuli to individuals in ant colonies. Smell plays a major role in ant communication, therefore it is important to control the chemicals displayed on the dummy using an inert coating. To this end, we use a gold-coated cuboid magnet. Gold film also prevents oxidation of the magnetic material inside the warm and humid nest, and improves motion by reducing friction. The antennae are fabricated from the tip of synthetic paint brush fibres.

The control system, shown in Figure 2.2b, consists of two permanent magnets to move the dummy and two electromagnetic coils to oscillate the dummy. It houses the cables and a sliding contact to follow the infinite rotation of the DC stage it is mounted on. The signal generator and amplifier can be commanded via a serial interface, and are housed outside the incubator of the tracking system. The adapter is adjustable in all relevant dimensions in order to center the magnetic capture with respect to the rotational movement, adjust the distance to the arena plane, and position the electromagnetic coils with respect to the guidance magnet. The adapter is 3D printed with a Form2TMusing clear resinTM.

The control system is mounted on a translation - translation - rotation (TTR) DC stage with integrated encoders and leadscrew/toothed belt transmissions. The maximal omnidirectional speed is 20 mm/sec and 675 °/sec.


(a) ©Bart Zijlstra, UNIL



(b) ©Bart Zijlstra, UNIL

Figure 2.2: Antbot (a) Magnetically actuated robotic dummy with mock antennae for tactile stimulus and worker of *L. acervorum*, resting on transparent Teflon[™] covered surface. Cuboid length is 2.5 mm (b) Adjustable control system with two permanent magnets for navigation and two electromagnetic coils for oscillating the antenna. The control system is mounted on a rotary stage. Permanent magnet diameter is 2 mm.

To ensure that the dummy is always under the control of the magnetic system, it must be navigated using a *unicycle* model, i.e. with blocked lateral movements:

$$\dot{\mathbf{x}} \approx v[\cos(\theta), \sin(\theta)]^T, \quad v \in [-v_{max}, v_{max}],$$
(2.1)

$$\dot{\theta} = \omega, \quad \omega \in [-\omega_{max}, \omega_{max}],$$
(2.2)

where x and θ are the position and the heading of the robotic dummy and v and ω the translation and rotation velocities $([\cdot]^T$ denotes the transpose). There are limits on the angular, ω_{max} , and lateral, v_{max} , speed determined by the hardware and safety considerations. To circumvent the lag effect introduced by the unavoidable distance between the guidance magnet and the dummy, it is beneficial to use a *forward-moving* unicycle model with $v \in [0, v_{max}]$. In autonomous navigation, this will also ensure that the dummy arrives head-first at the manipulation site.

actuator	range	Vmax	precision (bi-/unidirectional)	resolution	dimensions
PITMVT-80 linear table, DC-motor, lead screw PITMDT-34 rotary actuator, DC-motor with encoder	100mm	20 <i>mm</i> / <i>sec</i> 675 ⁰ / <i>sec</i>	$0.81 \pm 10 \mu m$ $7001 \pm 1400 \mu r a d$	0.5μm 700μr <i>ad</i>	243 × 83.5 × 25 <i>mm</i> 58 × 34 × 18.5 <i>mm</i>
motion controller			channels	Interface	API
PITMC-884.4DC			4	USB (RS-232 or TCP/IP)	PI TMGCS/PIPython
magnetic adaptor	guidance magnet	coils	signal generator	amplifier	IdA
guidance and signal actuation	$2 \times$ HKCM TM Z02x10Ni-N50, NdFeB, 1.4T, $2(\emptyset) \times 10(h)mm$	2× Coilcraft™RFS1317-824KL, removed shielding, 820μH	R&S TM HMF 2525 1 ch, 25 MHz	TOEтм7610 4Q, ±150 <i>W</i> , ±15 <i>A</i>	R&S Visa TM /RSInstruments python library
dummy and arena	targeted perception	dummy body	antennae	arena coating	arena dimensions
stimulus delivery	tactile	HKCM TM Q01x00.5x02.5Au-N52 NdFeB, gold coated, 1.43T, 2.5 × 1 × 0.5mm	Lascaux TM LR 15060 paintbrush Interlon TM synthetic fiber	CS Hyde ^{TM2} mil. FEP with Silicone adhesive, "Teflon sheet'	$145 \times 105 mm$
computer visual tracking			resolution [px]	detection precision	[xd/um] t
FORT system			$7920 \times 6004 = 48 M px$	0.02, calculated for 150	× 150mm FOV
assembly			precision direct (μ/σ)	precision magnetically guided (μ/σ)	total dimensions
stacked T-T-R manipulator + adaptor + tethered dum	my		0.02/0.01mm (N=50)	0.11/0.07 mm (N=50)	$250 \times 150 \times 120 \text{ mm}$

Table 2.1: Hardware configuration



Figure 2.3: Antbot integrated in FORT. (a) Antbot stage controllers (b) Humidity and illumination controllers (c) Heater (d) Camera (e) Visible light and IR flashlight (f) Humidifier (g) Extraction fan. **Green inset**: Ant arena (i) with nest (height: 2mm), covered with infrared pass filter (h), entrance to the right (width: 2mm). Underneath arena: Magnetic adapter with adjustable geometry round inset). Adapter is mounted on θ -rotary stage (j) and linear x - ystages (k). **Round inset**: magnetically guided robotic dummy (small gold cuboid (2.5 × 1.5 × 0.5 mm) above guidance magnets (m) and coils (n).

2.4 Software

The software developed for the teleoperated and automated manipulation has two main parts: perceptive processing and action planning. The perceptive part takes the tracked positions as input, calculates individual and colony level parameters, called proxies, that are suitable to study collective behavior. A representative example is the community partition of the interaction network to describe the social structure of a colony. As another example, individual activity levels can be used to study division of labour. These results can be visualized in real-time to support the action planning of the human operator who applies teleoperated stimuli or it can be used in a closed-loop scenario, where the hardware is controlled autonomously. Although the software structure is specifically designed for the Antbot, it is ready to be adapted to control any type of hardware. This framework is presented in Section 2.4.2 as the brain of the Antbot, which is described in Section 2.4.3. Figure 2.4 conceptualized the overall integrated operational framework. Through a hermes server (see Section 2.1 and Figure 2.1), the FORT tracking system provides tracked positions in real time. This data is processed by FORTloop to be displayed in a customizable form to a human observer (see for example Figure 2.8) or to control the relevant hardware.

2.4.1 Processing of social proxies

The data available after post-processing of the recorded images includes trajectories that are corrected for the heading of the animal, and interactions between individuals ³. Table 2.2 is a summary of the proxies that are currently used to describe the social life in an ant colony along with the a note on whether they have been inferred automatically or manually.

³A project to process the visual input with respect to individual pose in addition to tag position is currently ongoing.



Figure 2.4: Schematic of the FORTloop and the Antbot control framework

name	definition	level	significance	inference technique	utomation	references	remarks
social maturity	position in interaction network		social structure	soft partitioning on interaction graph	yes ¹	Richardson et al. (2021); Kay et al. (2022)	
foraging score	times individual left the nest		division of labour	counting	yes ⁵	this thesis, Richardson et al. (2022)	generalizable to any site (e.g. trash pile)
guarding score	entrance oriented static pose inside the nest		division of labour	geometric from tracking data	yes ⁵	this thesis, Richardson et al. (2022)	
brood care score	times observed performing		division of labour	temporal sampling from video	ou	this thesis	requires brood detection
activity	speed > <i>p_{min}</i> % body length / sec	i/c	unspecific/cyclic activity	time signal processing	yes^2	Richardson et al. (2017)	
interaction network	centrality/degree/betweenness	ပ	social structure	calculated from interaction graph	yes ³	Stroeymeyt et al. (2018)	
site-visiting network	node similarity on spatial fidelity	i/c	social structure	calculated from spatial fiedlity graph	yes ⁴	Richardson et al. (2022); Kay et al. (2022)	
trophallaxis	supervised learning/heuristic on pose	i/c	social structure	pose estimation	possible		in development
(allo-) grooming	supervised learning/heuristic on pose	i/c	social structure	pose estimation	possible		in development
¹ https://c4science.cf	1/source/facet unil/						

https://csscence.cn/source/iace__unu/ 2 https://github.com/formicidae-tracker/data-analysis-templates/blob/main/trajectory_analysis.jpynb 3 https://github.com/formicidae-tracker/data-analysis-templates/blob/main/interaction_network.jpynb 4 https://github.com/formicidae-tracker/data-analysis-templates/blob/main/spatial_fidelity.ipynb 5 will be published in the supplementary meterial of the manuscript of Chapter 3

Table 2.2: Behavioral proxies commonly used to evaluate ant tracking data.

2.4.2 FORTloop: experiment automation

The operational concepts specifically developed for the Antbot are generalized to build a generic platform for easy design and prototyping of manipulative experiments. To this end, we provide a comprehensive library of examples and reduce boilerplate code as much as possible. *FORTLoop* is essentially a concurrent python library, standardizing experimental programs that process tracking data from the FORT tracking system in real time. The data can be aggregated in interaction or spatial networks, for example, to obtain individual social parameters such a social maturity (Richardson et al., 2021), foraging score, and activity. These parameters can then be used interactively to perform the manipulation experiment.

FORTLoop is composed of two components that are accessible. The Loop class is the main process that includes a *hermes* client polling the tracking frames and manages all subprocesses and distributes the data to its Hook class instances. The user can instantiate as many Hook objects as needed, and pass them to the Loop before running it. The Hook class has an abstract method called process_frame(frame) that receives a dictionary containing the timestamp in seconds since epoch, the frame coordinate range xmax / ymax and the positions, an array of dimension $n \times 6$ for the n tags detected in the frame (tag ID, x coordinate, y coordinate, orientation, zone, timestamp). Besides receiving real-time data from the tracking system, FORTloop can also be used in a *simulation* mode, where data is read from a prerecorded experiment or *dryrun* mode, where recorded and real-time tracked data are concatenated. This can be helpful to prototype an experiment which includes hardware. Listing 2.1 shows a simple use case. FORTloop is available open-source including the comprehensive code for the application examples given below.⁴

The main intention of *FORTloop* is to handle the data and control flow for the *Antbot* described below, which represents the most complex application to date. In addition, its interface is abstracted in a way that it can be adapted easily to other use cases. The code for the following examples can be found on the *FORTloop* repository.⁵

⁴https://c4science.ch/source/fortloop/

⁵https://c4science.ch/source/fortloop/browse/master/demos/

Live plotting is a basic visual representation of tracking data. Listing 2.1 shows the code for a minimal example, based on which live plots such as the ones shown in Figure 2.8 can be generated.

```
from fortloop import Loop
from fortloop.hooks.liveplot import LivePlot
liveplot = LivePlot(update_period_sec=1.0)
fortloop = Loop([liveplot])
fortloop.run('trackinghost')
# for simulation, i.e. reading from file:
fortloop.simulate('prerecorded.hermes')
```

Listing 2.1: Python code for *FORTloop* live plotting

Simple hardware controlled by *Raspberry Pi*. The platform uses a *Raspberry Pi* to operate any hardware such as a single-board computer, micro-controller with network capability, servo motor, and LEDs as a response to some condition met by the tracking data. As an illustrative example, Figure 2.5 (a) shows the prototype of a selective food source setup, where food is displayed to the ants through a hole on the floor. Food with varying nutritious content can be place in the grey trays on the revolving table. The servo motor can then be actuated to display the chosen food to a selected individual. With FORTloop, a Loop can be used to gather information about the social behavior of an individual. This can be the amount of time a worker spends outside the nest or its position in the interaction network. According to the experimental design, food with a high nutritious value can then be displayed for example to individuals who are not specialized foragers to see if the reward influences their behavior. The code would look similar to Listing 2.2. This example has a worker *Hook* which revolves the food source and a concurrent Hook for live plotting.



Figure 2.5: FORTloop automated and reactive hardware examples. **Top**: rotating food sources, steered by servo motor and Raspberry Pi computer to selectively expose different quality food. Red rectangle represents ant nest, black circle is rotary table holding food trays that are presented through a hole in the ground. FORTloop program running on PC controls presented food source with respect to individual ID present in green polygonal zone. See code example in Listing 2.2. **Center**: moving wall nest, introduced in Chapter 1. Connecting path between two nest chambers can be extended with the underfloor linear DC motor. **Bottom**: 'ANT°C' system introduced in (Uslu, 2022; McGregor, 2022) as temperature regulation system for the ant nest using Thermo-Electric Cooling (TEC) elements. Temperature can be changed locally e.g. as response to aggregation of brood. FORTloop runs on a separate PC connected to the LAN.

```
1 import socket
2 from fortloop import Loop, Hook
3 from fortloop.hooks.liveplot import LivePlot
4 . . .
5
6 # Derive Hook class
7 class RPiServoHook(Hook):
      def __init__(self, rpi_ip, zone, focal_antid, ...):
8
          self.host = rpi_ip
9
          self.zone = Polygon(zone)
10
          self.focal_antid = focal_antid
11
12
          . . .
13
     def connect(self):
14
          self.s = socket.socket(...).connect()
15
16
          . . .
17
    def process_frame(self, frame):
18
          crds = frame["positions"][antid == self.focal_antid]
19
          # rotate food source if ant goes inside food zone
20
          if self.zone.contains(Point(crds[0, 1:3])):
21
              self.s.send("rotate positive")
22
          # and back when it leaves zone
23
          elif not self.zone.contains(Point(crds[0, 1:3])):
24
              self.s.send("rotate negative")
25
26
27 liveplot = LivePlot(update_period_sec=1.0)
28 rpiservo = RPiServoHook("130.xx.xx", zone, 'ant_of_interest')
29 fortloop = Loop([rpiservo, liveplot])
30 rpiservo.connect()
31 fortloop.run('trackinghost')
```

Listing 2.2: Python code for *FORTloop* food source example

Complex hardware with dedicated serial controllers. More complex hardware that comes with a serial controller should be integrated directly into a Loop. The *Ant Nest Temperature Controller system* (ANT°C) is a system that allows to change the floor temperature of the ant arena to provoke a collective reaction, such as the displacement of brood to a milder spot. The system has been developed by Fazil Uslu of the Microbiorobotic Laboratory and Sean McGregor of the Keller group (Uslu, 2022; McGregor, 2022). It consists of array of Peltier elements which are controlled by a Thermo-electric Cooler (TEC) Controller. The controller has a serial interface which, in the ANT°C system, is used in a Loop to command a change in temperature if a certain percentage of individuals aggregate in a given spot. Another example is the moving wall setup that is programmed to extend the access path as a response to activity. A motorized stage was used to insert and retract a mobile wall inside the nest. All these

hardware can be operated by the FORTloop framework.

2.4.3 Autonomous interactive experiments

Depending on the hypothesis and experimental design of a manipulative study, the Antbot must deliver the stimuli to individuals of different groups with sufficient *replicates* to obtain statistically significant results. These groups represent the set of *explanatory factors* and can be formed based on a temporally fixed features such as age and caste, or a variable feature like the current nest area and the activity state of the individuals within. At the same time, *randomization* might be necessary to average out systematic effects, e.g. effects coming from the time of the day when the manipulation is performed. The sum of these requirements, paired with the delicacy of applying the stimuli, poses a great challenge for action planning.

In the teleoperated experiment presented in Chapter 3, the operator guided the dummy to an individual during a predefined period in a predefined density area of the nest because it was expected that ant density would influence the *response*. A density-map based on real-time data was displayed as a visual aid, in order to follow the predefined experimental schedule. This way, more than 600 stimuli were applied in three different colonies to derive a statistically valid conclusion about the factors that influence the individual response. A lesson learned from this experiment is that choosing the next individual in the correct zone, safely navigating the dummy towards the target ant, and triggering the tactile signal at the correct distance from the target ant within the predefined time period is very demanding. Yet, this manipulation task has a comparatively simple experimental design. The analysis has shown that, among the approximately 100 individuals per colony, few were involved in sufficiently many interactions in different groups, e.g. in high and low density areas. Thus, the replication on the individual level was limited.

Programming the behavior of the robot for a diverse set of tasks in a complex and unpredictable environment demands reactive planning. The Antbot uses a *behavior tree* (BT), a technical alternative to a finite state automaton (FSA), for the behavior to implement experimental plans. BTs are more modular than FSAs, and describe the hierarchy of behavior in sub-trees instead of interlinked states. When the behavior has to be reprogrammed, which is anticipated, BTs do not suffer from the increased combinatorial complexity of rewiring FSAs. Equivalent to FSA, BT represents a procedure of graphic programming that streamlines the design of robot behavior, and facilitate cooperative development by experts from Biology and Robotics. The detailed description of behavior trees and common design paradigms can be found in (Colledanchise and Ögren, 2018).



Figure 2.6: Antbot behavior tree

Figure 2.6 shows the basic BT for an Antbot experiment. As the name suggests, the control flow is defined as a tree of subordinate behaviors. A given BT sub-tree is executed when it receives a *tick* emanated from the tree root and reaching its parent. Child behaviors receive the tick as they are ordered from left to right. Unlike FSAs, ticks are distributed periodically. In the Antbot, a tick is emitted whenever a new frame of tracking data is available. BTs consist of five main components:

- Execution nodes (leaves) consisting of action and condition
- Control flow nodes consisting of sequence, selector or fallback and parallel

In the program of Figure 2.6, the tick is emitted from the root which is the 'interaction sequence'

Chapter 2.	Robotic	manij	pulation	platform
------------	---------	-------	----------	----------

node	if <> success	else if <> running	else (if <> failure)
sequence	all	any*	any*
parallel	all	all	any
selector/fallback	any*	any*	all
return	success	running	failure

Table 2.3: Behavior tree control flow nodes. State returned based on child node states. *: priority in child tick execution order 'left to right', see Figure 2.6.

node, down the subordinate routines of either attempting an interaction or choosing a new target. Upon receiving a tick, each node returns either *success, running* or *failure*, based on its own state for leaves or the sub-tree status for control flow nodes. Table 2.3 summarizes the returned state of the different control flow nodes, based on the state of their children. A representative Antbot BT program is shown in Figure 2.6. The intent of this program is to automate the following experimental sequence that is followed in Chapter 3 to teleoperate the dummy.

- 1. safely navigate in front of a target individual
- 2. enter its safety perimeter from the front with the proper heading
- 3. apply tactile signal
- 4. move to the waiting area upon completion of the task
- 5. repeat the sequence every 5 min

Execution of even this relatively simple program requires reactions. First, the dummy may get stuck during navigation, e.g. on a piece of debris, and thus cannot be steered by the magnetic control system. Second, the allocated time budget to contact an individual may expire when the target ant is hard to reach. When such problems occur during teleoperation, the experimenter rectifies the situation by either recovering the dummy with the manipulator or changing the target. With the BT framework, a similar solution is achieved through navigation sub-trees that ensure that the dummy is magnetically guided to the control system at all times and time constraints are respected. The latter is a BT idiom with a selector and a condition

('timeout condition') that is in the sub-tree of a parallel node, together with an action that controls the condition ('timeout action' - 'timeout condition'). This is called an *eternal guard*⁶ as it keeps constantly checking for a set of conditions while a given action is performed (here the 'interact, evacuate' subtree). This way, an ant which seems unreachable, does not become a bottleneck for the continuation of the experiment. If it is desirable to insist on contacting a certain individual, the BT can be easily rewired accordingly, which illustrates the advantage of using this framework.



Figure 2.7: Antbot processes and data flow. Solid arrows: queues for asynchronous one-way communication. Solid boxes: Sub-processed synchronized by incoming tracking frames (FORTloop hooks). Dashed box: Asynchronous hardware control loop (FORTloop process) for trajectory generation and as a collision safety layer.

The BT control flow is purely sequential, yet, its actions might run in different threads. This is the case for the Antbot software that builds on the multithreaded FORTloop framework with multiple hooks and processes for objective generation, general control, navigation and low level motion planner (Figure 2.7). A single lightweight hook is used to generate the behavior⁷ and manage the control of other hooks by means of asynchronous inter-thread signal queues. These queues are also used for inter-hook (= inter-thread) exchange of data and states. In FORTloop programs, the main thread, which is launched initially when calling Loop.run(),

⁶*py_trees* package documentation: https://py-trees.readthedocs.io/

⁷The Antobot uses the *py_trees* BT package, release 2.1.6: https://github.com/splintered-reality/py_trees/

synchronizes the hook processes indirectly by distributing the data from the tracking system. This mechanism is used here to distribute the BT Tick. The hardware controller is implemented as an independent routine to generate a smooth trajectory and enforce the non-holonomic unicycle constraint which is needed to keep the magnetic dummy in capture. To this end, a FORTloop.Process class is implemented to run a customizable loop() method in an infinite and asynchronous loop. FORTloop.Processes do not receive any tracking frame data directly. In the Antbot program, the navigation loop provides the necessary data via a queue to the controller process. This data consists of temporally valid safe space for navigation and an intermediate goal.

The details of the global path planning algorithm and how it is connected to the local controller are given in Chapter 4. The internal state machines that interconnect the navigation with the controller are given in the Appendix (Figures A.1, A.2). All hardware communication with the function generator and the stage controller is managed by the controller process (Figure 2.7). The target ant can be selected randomly. Or, targets could be selected based on a social parameter (data aggregation block in Figure 2.7, see Section 2.4.1) or considering the probability for success in reaching them that is captured by a navigation cost.

As mentioned above, FORTloop has three ways of providing input data streams. To stream data from a prerecorded experiment, Loop.simulate() is used. This is useful for code development and hardware testing, especially when visual feedback from the tracking system is needed in real-time. Loop.dryrun() is designed for another scenario where data is streamed from a prerecorded experiment. Compared to a pure simulation, here the loop is closed and the timing is realistic as it is based on incoming tracking frames over the LAN. The dryrun mode is useful as a last verification stage, before the robot is set to physically move inside an ant colony, in which case the data is streamed from an ongoing tracking experiment with Loop.run() and no recorded data is involved. The coordinates of the manipulation stage and the camera must be coordinated through a calibration process. We use a linear transformation that is fitted with least squares after performing a calibration random walk with n = 50 steps of the tagged dummy over the entire arena. The details for performing the calibration are given

in Appendix A.2.3. The precision and repeatability of manipulation is reported in Table 2.1.

2.5 Experimental validation

Figure 2.2a shows the robotic dummy interacting with a *Leptothorax acervorum* worker. This dummy, which is designed to deliver a tactile stimulus, is used throughout the entire thesis. In this section, the concrete steps that were taken to demonstrate the use of Antbot in behavioral experiments is discussed. Here, we describe a study that involves three colonies of *L. acervorum* and approximately 600 targeted interactions with a dummy as an extension to the basic manipulation scheme introduced in Section 2.2. In Chapter 3, the data collected throughout this study is analyzed using statistical tools to gain insights into the factors that drive individual aggression. In this section, the focus is on the technical aspects.

As ants must be kept immobile to glue the tag markers for tracking, which is challenging to perform with relatively small ants such as *L. acervorum*. Ants are temporarily anesthetized either by releasing CO_2 or cooling them on ice. Application of these techniques requires extensive training, in order to acquire skills that are required to reduce tag loss due to imprecise application of the glue or tag, and to avoid increased mortality due to keeping the animals anesthetized for too long. A new technique was invented to tag the ants by immobilizing them mechanically and without anesthesia, allowing to work without time pressure (Appendix A.2).

The fabrication and insertion of the robot is explained in Appendix A.2.2. The entire protocol to set up an Antbot experiment, involving the referencing and calibration of the manipulator stage is given in Appendix A.2.3. The *keyboardAntbot* program used to command the robotic dummy for teleoperation and is explained in Appendix A.2.3 as well.

The nest arena is placed inside the tracking system and left for acclimatization for at least two days. Using a second nest entrance, hidden to the ants, the robot dummy is inserted into the colony in a minimally invasive fashion (Figure 2.8). The set goal is to contact individuals in different density zones, where these density zones are computed in real time using the tracking data. FORTloop displays the density plot to allow the operator to select individuals

Chapter 2. Robotic manipulation platform

based on their location in the updated map. Using the visual feedback from the video stream plus the density map, the operator is then able to steer the dummy to an individual that is chosen based on the density zone dictated by the experimental plan (compare Table 3.1). Teleoperation works smoothly in most cases. Exceptions include dummy loss due to debris collected during navigation and mistakes made by the operator ('Problem' group in in Figure 2.9a).



Figure 2.8: Data visualization as visual aid for experimenter for Antbot teleoperation. Left: FORT live videostream. Red circle: dummy. Red lines left: hidden door for dummy exchange. Red below: main nest entrance. Right: Density plot using FORTloop (compare similar code example in Listing 2.1). Gaussian kernel estimate and two equidistant contour curves. Green pentagon: dummy location.

The dummy is able to provoke a spectrum of behavioral responses in addition to aggression (Figure 2.9). To investigate the effectiveness of a targeted actuated tactile interaction of the dummy, we compared interactions of ants with a dummy that was kept stationary and with the same dummy that was navigated and actuated. Interactions between an idle and an active individual are naturally directed: the active individual is contacting the inactive one. Therefore it is expected that indifferent responses are unlikely for individuals that interact with the static dummy, as the fact that the interaction takes place presumes that the active ant is not indifferent towards the dummy. Figure 2.9a compares the behavioral response of individuals to targeted robot interactions to when the static robot is encountered. Ants almost never get alarmed when they encounter the stationary dummy. Furthermore, we analyzed the duration of the interaction during exploratory interactions when the ant was previously active



and found no significant difference (Figure 2.9b).

Figure 2.9: (a): Behavioral response of individuals to targeted robot interaction (moving) or when static robot is encountered. Problem represents mismanipulations of the operator and few situations where stick slip motion led to a harsh bump. (b): Duration of exploratory interactions are not significantly different when robot is static or moving (M.W.U.-test: U = 45063.5, p = 0.138).

Chemical cues detected by the antennae are central for ant to ant communication. A colony specific smell is used to discriminate nest mates from intruders and other foreign objects. Cuticular hydrocarbons (CHCs) act as a pheromone for recognition. Hydrocarbons passively deposited on other ants or objects can serve as recognition signal (Hölldobler et al., 1990). To avoid the formation of any kind of foreign smell on the dummy, we used a gold-coated magnet that was washed thoroughly before each experiment. The same tag and glue that was used for the ants are used to tag the magnet (see Appendix A.2.3).

In the beginning of each experiment, a dummy was placed in the center of the nest for seven consecutive days. After this acclimatization phase, the dummy was used in the following three days for interactions with the ants. The dummy was removed using the hidden passage, and placed in Hexane for ten minutes to extract the chemical surface profile. The chemical surface profiles of the dummies and a worker ant from the respective colony were analyzed after each experiment. The profiles were detected with a gas chromatography-mass spectrometry (GCMS) machine from Thermo Scientific[™]and the locations, intensity and spectra of the CHC peaks were assessed with the Xcalibur[™]1.4 SR1 software from Thermo Scientific[™](ISIS peak detection algorithm, NIST spectra library). Three hydrocarbon molecules, one of which

 $(C_{27}H_{56})$ is reported in the CHC profile in species of the *Myrmicinae* subfamiliy (Sprenger and Menzel, 2020), were identified in the chromatogram (Figure 2.10). For all molecules, the intensity was significantly higher in the workers than in the fresh dummies without acclimation $(C_{34}H_{70}: 18.9 \times, p < 10^{-24}, C_{27}H_{56}: 10.7 \times, p < 10^{-19}, C_{27}H_{56}: 4.8 \times, p < 10^{-12}, C_{44}H_{90}: 1.6 \times, p < 10^{-8}$, area under peak, t-test). In the profiles of the acclimated dummies, the average intensity was lower compared to that of the worker ants but still significantly higher than for peaks in the profiles of the fresh dummies $(C_{34}H_{70}: 4.6 \times, p < 10^{-12}, C_{27}H_{56}: 4.75 \times, p < 10^{-12}, C_{27}H_{56}: 2.1 \times, p < 0.005, C_{44}H_{90}: 1.2 \times, p = 0.002$, area under peak, t-test). We did not find a significantly different behavioral response to dummies that are freshly used or acclimatized inside the nest (Figure 2.11, see also Chapter 3).

2.5.1 Towards autonomous manipulation

We performed a test using the Antbot behavior defined in Figure 2.6 to demonstrate the potential for automated manipulation inside an ant colony. Figures 2.12 and 2.13 show snapshots from a video sequence of an experimental sequence performed with a physical dummy inside the arena using prerecorded experimental data and the 'dryrun' mode of FORTloop. This helps to spot current problems that will need to be addressed in the future in order to achieve a fully autonomous experiment that runs stable over the entire duration of the experiment (\approx 7 days). In this regard, two major challenges are identified. First, the robotic dummy sometimes fails to retract after the interaction is complete. This is caused by the security layer of the controller which detects a collision and stops all movements (Figure 2.7). The controller essentially ensures that the assumptions in terms of admissible collisions for the navigation algorithm are met (see Chapter 4). Interactions, which naturally imply with very close contact, and other situations robotic dummy is blocked by the ants occur frequently and can impede to pursue an experimental schedule. A solution for this problem is therefore necessary and could be approached by (1) disabling the collision detection after interactions or if stuck for long enough, (2) allowing the otherwise unidirectional unicycle to move in both directions (in this solution, it must be ensured that the robotic dummy will return to the



Figure 2.10: GCMS Chromatograms of worker and dummy chemical surface profiles for different colonies.



Figure 2.11: Individual behavioral with respect to dummy acclimatization.

forward-moving scheme as soon as possible in order to arrive head-first at the next interaction site) or (3) implementing a tailored unblocking behavior using an alternative motion planning approach to seek the closest free robot configuration. All solutions can easily be integrated an flexibly tested with the BT that governs the robot behavior. Second, in Figure 2.13 the robotic dummy can be seen to perform 'pirouette' movements, in which it does a full turn to finally continue in the same direction. This is presumably caused by a short discontinuity in the motion planning which induces a high angular speed causes it to perform a full rotation instead of reverting the movement due to the forward-moving constraint ($v \in [0, v_{max}]$ in Equation 2.1). The problem needs further investigation, but can potentially be solved either damping the angular velocity in the case of a drastic jump of the local planner reference or allowing reverse movements on a short term.



Figure 2.12: Dryrun demo, start configuration, continuation in Figure 2.13. Autonomous experiment sequence: navigate to target ant > align heading > perform interaction > navigate to safe space and wait until timeout. Antbot dummy is marked with black circle, its heading is indicated with green rectangle and the trajectory with green dots. Ants are marked with black dots and heading is indicated with red bar. Nest entrance is seen in the bottom. Secondary (closed) entrance on the left. Green text represents state changes in behavior tree of Figure 2.6.



Figure 2.13: Dryrun demo. See caption of Figure 2.12. Time evolves left to right, top to bottom. Green dots depict robotic dummy positions sampled at $\Delta t = 500$ ms.

Table 2.4 shows the performance evaluation of the integrated Antbot system. An overall latency of \approx 1300 ms for navigation and \approx 130 ms on the security layer (local motion planner receiving tracking frames at 8 Hz) where collisions are ultimately avoided. This is well withing the acceptable range of the values reported for safe navigation of human scale robots in unknown cluttered environments (Falanga et al., 2019) (as the Antbots' inertia is negligible, it can profit from the full acceleration of the stage). FORT is designed to maximize throughput of acquisition in observational experiments, and not yet optimized for real-time manipulation experiments. If the computational load of the host exceeds the capacity, frames are automatically dropped. For this reason, the delay of tracking is a theoretical upper bound. As an example, while tracking ants at 8 Hz, FORT drops up to 100 frames in a week-long experiment (i.e. dropping rate < 10⁻⁵), hence the upper bound for delay can be given as 125 ms. A list of improvements for the computational efficiency of bottlenecks in the navigation algorithm is given in Appendix A.1.2.

Module/component	latency [ms]	jitter _{RMS} [ms]
FORT tracking	125 (upper limit ¹)	10.9
Network ⁷	< 0.2	< 0.1
closed navigation loop ⁶	1057.9	97.1
pyantbot navigation	981.7	75.0
path planning	961.8	74.3
triangulation	426.4	44.9
graph construction ²	494.8	34.0
subregion identification ³	39.6	35.4
graph search ⁴	< 1.0	< 1.0
pyantbot local motion planner ⁵	2.0	0.3

Table 2.4: Antbot loop characteristics (N = 16838 frames $\hat{=} > 35$ min) at 8 Hz standard tracking frequency. Periods where measured with python's *time* package perf_counter function. ¹ FORT is not designed to respect RT constraints (explanation see text). 8 Hz is standard achievable single host tracking frequency, therefore the latency is an upper bound only. ² Derivation of the motion graph with edge cost according to distance and clearance, see Section 4.4.1. ³ Identification of subregion coverd by previous path solution needed for temporally persistent planning, see Section 4.4.2. ⁴ Shortest path algorithm on motion graph (using python *networkx* function single_source_dijkstra on full domain and subregion, see Section 4.4.1. ⁵ Smooth and safe low level trajectory generation in local free space, see Section 4.5.1. ⁶ hardware controller loop for smooth trajectory generation (see Figure 2.7) is limited by the local motion planner latency and set to $f_{ctrl} = 10$ Hz, processing the tracking frames at 8 Hz in the collision security layer. ⁷ Inferred with ping (N = 1000) and 5 kB packet size.

2.6 Conclusion

With the compact hardware in the form of a stacked TTR setup, using DC lead screw linear stages, a DC rotary stage and a custom designed magnetic manipulator to guide and actuate the ant-sized robotic dummy, the hardware requirements on mobility, workspace, size, accessibility, versatility are met by design (Requirements R1, R2, R3, R8). The electromagnetic coils allow to actuate the fine motion of the dummy to produce a tactile signal that can be varied instantaneously if needed. To this end, the signal generator can be controlled remotely and automated with software (Requirement R6). The software introduced here is written with the many experimental scenarios in mind that have already been used with ants. FORTloop allows to build automated experiments in a safe and fast way by hiding the complexity of process management and data acquisition from the user. This is to ensure that programs will run stable for the long experimental periods commonly used here (Requirement R8). In the behavioral experiment presented in this section, it was shown that, although the robotic dummy acquires potential CHC pheromones, this does not affect the behavioral response (Requirement R5). It is also capable to elicit a spectrum of behavioral response rather than a simple single reaction, such as aggression (Requirement R4). The Antbot is indeed capable of provoking reactions that depend on various measurable social factors, which will be investigated in detail in Chapter 3. To achieve full autonomy of the robotic dummy in an experiment, the action planning and navigation need to be automated (Requirements R9, R10). The former is addressed with the BT framework that allows to compose the behavior of the Antbot in a modular way, which is important in the prototyping process of a specific experiment. The latter capability is addressed separately in Chapter 4, as the problem of navigation in a crowded environment represents a challenge also identified in the field of robotics. The algorithm developed therein was tested in a demonstrative experiment presented in this section, which represents the proof of the concept.

3 Tactile stimulation of individuals in an ant colony

The work reported in this chapter will be submitted as:

Robot-assisted manipulation to disentangle behavioral plasticity and task specialization as factors for aggression in ants Matthias Rüegg^{1,2*}, Alba Motes Rodrigo^{2*}, Thomas Owen Richardson^{2,3}, Nathalie Stroeymeyt^{2,3}, and Mahmut Selman Sakar¹, and Laurent Keller²

* These authors contributed equally.

¹ Institute of Mechanical Engineering, École polytechnique fédérale de Lausanne, Switzerland

² Department of Ecology and Evolution, Université de Lausanne, Switzerland

³ School of Biological Sciences, University of Bristol, UK

3.1 Introduction

Ants are one of the most successful taxa on the planet, surviving in almost every ecosystem on earth. This colonizing success is partly due to their behavioral variation and flexibility which can be found at different organizational levels, namely between individuals, castes and colonies (Jandt et al., 2014; Oster and Wilson, 1978). behavioral variation between castes is also referred to as division of labour, where different groups of individuals perform different tasks in the colony based on their age, genotype and morphology. Overall division of labour has been shown to improve group productivity as it allows individuals to specialize in a given task and reduce task-changing costs (Beshers and Fewell, 2001). In a colony, different castes are in charge of tasks such as reproduction, nursing the brood and queen, foraging, cleaning and defending the nest (Mersch et al., 2013; Richardson et al., 2022). Caste identity has been linked to specific behavioral phenotypes and intracolonial variance in aggression, broodcare and exploration (behaviors typically associated with caste identity) has been positively correlated with per worker productivity (Modlmeier and Foitzik, 2011; Modlmeier et al., 2012). For instance, *Myrmica ruginodis* foragers are bolder, more aggressive and more active than nurses caring for the queen and brood (Chapman et al., 2011). Similarly, slender ant (*Leptothorax acervorum*) workers that engage in less broodcare early in life are more aggressive when tested in individual behavioral assays with a dead non-nestmate (Kühbandner et al., 2014). Highly exploratory workers of this species were also found to be more aggressive than less exploratory individuals.

Guarding the nest entrances as a defensive strategy is commonly practiced by multiple ant species such as leaf-cutting ants (*Atta laevigata*, (Whitehouse and Jaffe, 1996)), slender ants (*Leptothorax acervorum*) and red wood ants (*Formica fusta*). Guarding individuals have been shown to display higher levels of aggression towards intra-nest parasitic myrmecophiles compared to foragers (Parmentier et al., 2015). In addition to ants performing specific tasks, some studies have reported a certain group of ants (in some cases up to 50% of the colony) that do not seem to contribute to colony labour (Charbonneau et al., 2015) named 'lazy ants'. Removal experiments revealed that lazy ants take over foraging and broodcaring tasks when specialised, active individuals die or are removed from the colony. They seem to act as a reserve to maintain the active workforce (Charbonneau et al., 2017). Furthermore, comparisons between wild and laboratory colonies of the ant *Temnothorax regulatus* have shown that lazy ants are not an artifact of laboratory conditions and that lazy ants, foragers and nurses occur in similar proportions across experimental settings (Charbonneau et al., 2015).

In addition to the division of labour between castes, there is also behavioral variation among members of a given caste. This intra-caste variation can be caused, for example, by individual ants' experiences and social context. Aggression towards intruders is also regulated by the social context. In the Argentine ant *Linepithema humile*, workers display higher levels of aggression towards non-nestmates when tested in a group compared to when tested alone. Furthermore, the presence of brood reduced aggression during dyadic encounters with non-nest-mates as the workers often engaged in broodcare and broodcarrying (Buczkowski and Silverman, 2005). To the best of our knowledge, no study to date has investigated how the immediate social context of interactions inside the nest influences individual behavioral responses.

In this study, we jointly investigated how individual task allocation, stimuli characteristics and social context (measured as intra-nest density) relate to individual behavioral responses and shape intracolony behavioral variation of the slender ant *L. acervorum. L. acervorum* is a monomorphic, facultatively polygynous ant with colonies ranging from 50 to a few hundred workers. In this species, both the reproductive queen and the workers lay eggs (Bourke, 1991). To maximize stimuli homogeneity and control the context of the ant-stimuli interactions, we used a teleoperated dummy actuated by the Antbot system introduced in Chapter 2. This methodology allowed us to evaluate the behavioral responses of individual ants towards a novel stimulus within their colony of origin and in a nest setting, recreating the ant's natural social environment. Furthermore, we could select the location of the interaction between the ant and the dummy, allowing us to evaluate how intra-nest density influences ants' behavioral responses.

3.2 Methods

3.2.1 Robotic manipulation system

The robotic setup consists of a magnetically guided, exchangeable dummy and an external manipulator to navigate the dummy among the ants and generate a precise mechanical stimulus (simulated antennation, Figure 3.1, top right). The dummy was fabricated according to Section 2.3 and Appendix A.2.2. Each dummy was only used with one colony. The manipulator system is to guide the dummy is described Section 2.3. Here we used the system in teleoperation via the *keyboardAntbot* program with visual feedback from a live video stream and a real-time density map (Section 2.5, Figure 2.8). The details of the system are explained in Chapter 2.

3.2.2 Experimental set-up

Three *Leptothorax acervorum* colonies consisting of a queen and approximately 100 workers were collected on August 13, 2020 in *Anzeindaz, Switzerland*. The colonies were housed in TeflonTMcovered plastic boxes that allowed the dummy to slide in the nest during the experiments. The walls of the boxes were covered with FluonTM, an extremely smooth substance that prevent ants from climbing out of the boxes. The boxes were equipped with a water tube and a 75 × 52 × 2 mm nest made of cardboard and covered with an infrared-pass filter glass opaque for visible light. The ants were fed once a week with diluted sugar water, *Drosophila* flies and honey jelly. At the time of the experiments the colonies had been in the laboratory for approximately six months. Ten days before the experimental manipulation, the workers were tagged using unique *ARTags* markers of 0.7 mm (Figure 3.1 top right, (Mersch et al., 2013)) and moved to a clean nest. The tagging was done using an immobilization technique that does not require the use of any form of anesthetic or cooling (Appendix A.2.1). The tags were glued using fast-drying glue (Pattex ultraTM). Before the start of the experimental manipulations, the dummy was also tagged. To track the ants, we used the Formicidae Tracking System (FORT)¹, an automated marker-based tracking system that records individual ant trajectories

¹https://github.com/formicidae-tracker/

over extended periods of time and produces a live video stream and real-time data to operate the robot. A daily climate cycle of 12 hours daylight with gradual dawn during 30 minutes and a temperature of 17°C at 70% humidity was maintained throughout the experiment.

3.2.3 Experimental design

For each colony, the experiment was divided into a seven-day acclimation phase, followed by a six-day experimental phase. The experimental phase consisted of two four-hour sessions per day on six consecutive days. Every session consisted of four one-hour periods, including two hours when targeted interactions were conducted (moving periods) and two hours when no targeted interactions were conducted and the dummy remained immobile (static periods, Table 3.1). We randomly selected half of the sessions to start with a moving period and the other half to start with a static period. A targeted interaction took place when the dummy entered in physical contact with the targeted ant. Each interaction consisted of a simulated antennation event where the dummy was directed towards an ant and performed a sinusoidal wiggling signal at 1 Hz for 15 seconds. Interactions were conducted every 5 minutes. In some cases (N = 51) interaction events had to be discarded due to technical problems. These instances involved the loss of the magnetic capture of the dummy, debris jammed under the magnet that prevented its movement or forceful crashes with ants after the dummy was stuck or the operator failed.

To avoid overly targeting easier-to-reach ants in less crowded zones of the nest, mobile periods when the dummy was teleoperated took place in predefined zones with varying ant densities. To define real-time density zones in the nest, the trajectory of every ant was retrieved in real-time from the FORT tracking system and plotted in an equidistant density contour plot overlaid on the live feed. This plot contained three delimited zones calculated using a Gaussian kernel density estimation that served as visual feedback for the experimenter to direct the dummy towards an ant in the desired density zone (Figure 3.1, top left). The density of the zone were the interactions of a given period took place (i.e., low, middle or high) was defined randomly a priori so that each zone was targeted in four sessions during the experiment (see

	da	y 1	da	y 2	da	y 3	da	y 4	da	y 5	da	y 6
morning	low density	m s m s	mid. dens.	m s m s	low density	m s m s	low density	m s m s	mid. dens.	m s m s	low density	m s m s
afternoon	mid. dens.	m s m s	high dens.	m s m s	high dens.	m s m s	mid. dens.	m s m s	high dens.	m s m s	high dens.	m s m s
	a	cclin	natiz	ed dı	ımm	acclimatized dummy						hd.

Table 3.1: Experimental plan. Each day included two sessions where dummy interactions were carried out in either high, middle or low density areas with four periods each during which the dummy was moving (\mathbf{m}) or static (\mathbf{s}). Each period was one hour long.

Table 3.1). To investigate whether the dummy could potentially acquire a chemical profile while inside the nest that influenced the ants' behavior, the first six experimental sessions were performed with an acclimatized dummy placed inside the nest seven days before the start of the first session whereas the remaining six sessions were each conducted with a freshly fabricated dummy. To avoid disturbing the colony, each dummy was first introduced into the nest through a secondary nest opening that remained closed otherwise rather than via the main nest entrance.

3.2.4 Data collection and coding

Interactions between ants and the dummy were detected with the FORT-myrmidon postprocessing library.² Two ellipses were defined for each individual, one comprising the head and antenna region and the other comprising the body. Interactions were detected when the head ellipse of a given individual collided with the head or body ellipses of another individual or the dummy. The dummy was defined by a single ellipse.

Fifty interactions between ants and the dummy during immobile periods were randomly selected to evaluate differences in response towards moving and immobile dummies. In

²https://github.com/formicidae-tracker/myrmidon

a preliminary inspection of the video recordings from the periods where the dummy was teleoperated (moving periods), five distinctive behavioral responses towards the dummy were identified (Table 3.2). The recording of each interaction event between the dummy and the targeted ant during moving periods was cut to a one-minute video sequence, showing the interaction at second ten. Any video metadata that would allow identifying the time or colony, was masked. These video sequences where then shuffled and blind-coded by a person who did not take part in the data collection (AMR). In addition to the behavioral response towards the dummy, for each interaction during moving periods we recorded the density of ants in the zone where the interaction took place, the task that the focal ant was performing right before the interaction, the ant ID, the colony ID and whether the dummy was acclimatized or freshly-made.

In addition to interaction- and dummy-specific factors (e.g. zone density and whether the dummy was acclimatized or not), we also recorded some individual-level characteristics that are generally associated with division of labor in social insects. Specifically, we coded how much brood care, foraging and guarding each individual ant performed during the six days of the experiment. Individual brood care frequency was calculated by analyzing five-minute videos every four hours during the acclimation phase and determining which ants were doing brood care during the five-minute videos. The brood care frequency of each ant was then calculated as the sum of the number of five-minute videos in which the ant had been observed doing brood care. The foraging frequency of each ant was calculated as the number of times an individual left the nest, since all food sources were located outside of the nest (Charbonneau et al., 2015). Guarding time was calculated as the total amount of time an individual was observed within five body lengths (measured as the colony average) to the nest entrance, immobile (i.e. moving less than 5% of the body length per second) and oriented within $\pm 45^{o}$ of the nest entrance (see also (Richardson et al., 2022)).

3.2.5 Statistical analyses

We conducted the statistical analyses in *R* (version 3.6.3; R Core Team, 2020) and *RStudio* (version 1.2.5031; RStudio Team, 2020). We used Generalized Linear Mixed-Effects model (GLMMs) to evaluate the effects of different predictors on three main response variables:

- Model 1: the probability of ants to explore (antennate) the dummy
- Model 3: the probability of ants to respond to the dummy after a targeted interaction
- Models 2 and 4: the probability of ants to respond alarmed towards the dummy after a targeted interaction

We defined alarmed responses as any response that included biting, excited bouts or both (see Table 3.2 for definitions). Climbing on top of the dummy was not classified as part of the alarmed response because it was not obviously related with behaviors associated with anxiety or agitation.

In Model 1 we evaluated whether ants were more likely to explore a moving dummy compared to a static dummy. In Model 2 we evaluated whether ants were more likely to respond alarmed to a moving compared to a static dummy. Models 1 and 2 were GLMMs with binomial error structures. The response variables were whether the ants explored the dummy after an interaction (yes/no, Model 1) and whether they responded alarmed to the dummy after an interaction (yes/no, Model 2). As predictor, we considered the state of the dummy (moving/static). We used a subset of the data to fit Models 1 and 2 that only comprised 'active' ants ($N_{ants} = 178$), as these were the only ants that interacted with the non-moving dummy (targeted interactions with the moving dummy involved both active and non-active ants). Active ants were classified as those engaging in antennation, walking or brood care at the time of the interaction with the dummy. For all subsequent statistical analyses we only considered the targeted interactions performed with the moving dummy ($N_{interactions} = 651$) as these where conducted under the control of the experimenter and were balanced across experimental conditions and colonies.

In Model 3, we investigated how the probability of ants to respond towards the dummy (binary variable) was influenced by:

- ant density in the zone where the interaction took place (factor with three levels: high, medium or low)
- 2) task performed right before the interaction
- 3) dummy acclimatization (factor with two levels: acclimatized or freshly-made)
- 4) brood care frequency
- 5) foraging frequency
- 6) guarding time

The task variable was a factor with five levels reflecting the behavior that the targeted ant was doing right before the dummy interacted with it. These behaviors included brood care, being frozen (immobile with no antenna movement), being idle (immobile with antenna movement), antennating with another ant or the dummy and walking. Given that 56% of individuals were targeted while doing multiple tasks, the random slope of task within individual could have been included as a random effect in the model. However, the inclusion of this random slope caused convergence and estimation problems as the slopes of some factor levels within individual ID where unidentifiable. Consequently, we opted to not include this random slope in the model. This model had a binomial error distribution and a logic function. Model 4 had the same fixed and random effect structure as Model 3 but this time we evaluated the predictors' influence on the probability of the ants to respond alarmed to the dummy. Confidence intervals of model estimates were calculated via parametric bootstrapping, where new instances of the response were generated based on model results and the model refitted to these new instances allowing us to derive estimate distributions.

Before fitting Models 3 and 4 described above, we z-transformed covariates (brood care frequency, foraging frequency and guarding time) to have a mean of zero and standard deviation of one to facilitate the interpretation of the coefficient estimates (Schielzeth, 2010). Models 1 to 4 included the random intercept of individual ID as well as colony ID as control predictor. All models were checked for col-linearity and overall stability without finding any issues. We drew inference by comparing the full models containing all fixed and random predictors with a reduced null model lacking the predictors of interest but containing all other model elements (Forstmeier and Schielzeth, 2011) using a likelihood ratio test ('Chisq' in R function ANOVA). This approach of initially testing the significance of all test predictors prevents what is known as 'cryptic multiple testing' and maintains type 1 error rates at the nominal level of 0.05 (Forstmeier and Schielzeth, 2011). P-values for fixed effects were obtained using the R function drop1. Pairwise comparisons were conducted using the function glmer from the package with the same name. All models were fitted with the function glmer from the results obtained from Models 3 and 4 using the function mcNemar from the package *epibasix*.
3.3 Results

In total, we coded 801 interactions between the dummy and the individual ants (Figure 3.1). Of these, 702 were interactions where the dummy was mobile (i.e., teleoperated to interact with an ant). We excluded all interactions where there was a technical problem (N = 51, see Methods), leaving a total of 752 interactions with 208 different ants. Most ants (77%) were targeted only once during the experiments. We observed five different types of responses of the ants towards the dummy: indifference, antennation, excited bouts, climbing on top, and biting (see Table 3.2). The frequencies of each response as well as the observed combinations of responses are shown in Figure 3.1.

Ants were significantly more likely to respond alarmed towards the dummy when it was moving than when it was kept stationary (Model 1: $\chi^2 = 60.31$, df = 1, p < 0.001, estimate \pm SD = 9.33 ± 1.73). However, ants were less likely to explore the dummy when it was moving than when it was stationary (Model 2: $\chi^2 = 35.01$, df = 1, p < 0.001, estimate \pm SD = -2.72 ± 0.63).

3.3.1 Factors influencing the probability of responding to the dummy

We evaluated how the probability of ants responding to the moving dummy was influenced by the density of workers in the zone of contact, acclimatization of the dummy (i.e. whether or not it had been in the colony during the six days prior to the experiment), the task performed just before the interaction, brood care frequency, foraging frequency and time spent guarding. The comparison of the full and null models revealed that the predictors were significant as a whole ($\chi^2 = 68.73$, df = 10, p < 0.001, Figure 3.2). Regarding individual fixed effects, we found that the density of ants in the zone where the interaction took place had a significant effect on the probability that ants responded to the dummy (Likelihood ratio test = 40.75, df = 2, p < 0.001). Similarly, the task that the ants were performing just before interacting with the dummy significantly influenced the probability of response (Likelihood ratio test = 27.32, df = 4, p < 0.001). Ants' probability of response was not significantly affected by the acclimatization of the dummy (Likelihood ratio test = 2.04, df = 1, p = 0.15), the individual brood care frequency

<i>Indifferent</i> : The ant is touched by the wiggling dummy but shows no response (i.e., the head does not orient towards the dummy and occasionally the ant just walks/runs away without turning towards the dummy).
<i>Exploratory</i> : The ant touches the dummy with the antennae.
<i>Excited bouts</i> : After antenating the dummy the ant quickly moves away from it and proceeds to interact in quick succession with ants nearby or to check on eggs. Occasionally the ant will return to interact again with the dummy.
<i>Climbing on top</i> : The ant climbs on top of the dummy placing head and thorax on top of it.
<i>Biting</i> : Ant bites the dummy's mock antennae.

Table 3.2: Ethogram describing the different behavioral responses of the ants towards the dummy.



Figure 3.1: Top left panel illustrates the three density zones and their limits inside of the nest. Warmer colors represent high density zones and colder colors represent lower densities. Each number in blue represents an ant and the green symbol marks the position of the dummy. The double arrow marks the nest entrance. The top right picture illustrates a tagged ant antennating with the tagged dummy. The bottom graph depicts the frequencies of the different behavioral responses observed (left) and the different behavioral combinations observed (top).

(Likelihood ratio test = 0.085, p = 0.77), the individual foraging frequency (Likelihood ratio test = 0.057, p = 0.81) or the guarding time (Likelihood ratio test = 1.46, p = 0.23).

Ants were less likely to respond if the interaction took place in a high-density zone than a middle- or low-density zone (high-low: odds ratio = 0.19, p < 0.001; high-middle: odds odds ratio = 0.23, p < 0.001; middle-low: odds ratio = 0.82, p = 0.83). To exclude the possibility that the observed differences in the response could stem from intrinsic differences in response thresholds between ants located in the different density zones, we compared the the responses of ants that are stimulated at least once in both the high- and middle-density zones or in the high- and low-density zones ($N_{ants} = 53$). The probability of responding to the dummy for a given ant was significantly lower when ants were in the high-density zone than in the middleor low-density zones (Table 3.3). The same results were obtained when only a single value per ant and density zone was considered (Table 3.3).

Density zone	Sample per ant/task	Nants	McNemar- χ^2	Odds ratio	95%CI	р
High-Mid	all observations	53	37.31	7.37	3.89, 28.16	<0.001
High-Low	all observations	53	41.34	5.43	3.24, 12.31	<0.001
High-Mid	one observation one observation	53	21.02	7.0	3.17, 98.89	<0.001
High-Low		53	17.35	5.0	2.43, 22.78	<0.001

Table 3.3: Pairwise comparisons of the probability to respond towards the dummy of ants tested in two density zones.

The observed differences in probability of response towards the dummy could be due to the intrinsic differences in response thresholds among the ants performing different tasks. To test this hypothesis, we reevaluated the pairwise differences between tasks, considering only those individuals that had been tested when performing both tasks. Three of the four previously significant pairwise comparisons were found to be significant both when all observations per ant and task were used, and when only one observation per ant and task was considered (Table 3.4). There was no significant difference between ants performing antennation and brood care when only considering those individuals that were tested when performing both tasks (N = 6).

To evaluate whether being in physical contact with the brood influenced the probability to

behaviors	Sample per ant/task	Nants	McNemar- χ^2	Odds ratio	95%CI	р
Antennation-	all observations	13	2.11	×	×	0.146
Brood care						
Frozen-Brood	all observations	37	7.87	2.29	1.31, 4.74	0.005
Idle-Brood	all observations	12	9.09	×	×	0.003
Idle-Frozen	all observations	46	98.43	36.33	15.35, -132.92	< 0.001
Antennation-	one observation	13	4.08	×	×	0.043
Brood care						
Frozen-Brood	one observation	37	6.91	2.7	1.34, 7.80	0.009
Idle-Brood	one observation	12	4.17	×	×	0.04
Idle-Frozen	one observation	46	23.36	11	4.33, -48.98	< 0.001

Table 3.4: Pairwise comparisons of the probability to respond towards the dummy of ants tested when doing multiple tasks. (×): In those tests the odds ratio could not be calculated due to the number of discordant pairs being smaller than 30, the results should be taken with caution.

respond to the dummy, we compared the probability of response of ants that had been tested both when they were in physical contact with brood and when they were not (N = 55). Ants in the presence of brood were less likely to respond to the dummy (McNemar $\chi^2 = 72.75$, p < 0.001, odds ratio = 6.15, 95%CI[3.99, 11.57]) and this difference remained significant when only one response per ant in each condition (presence or absence of brood) was considered (McNemar $\chi^2 = 19.59$, p < 0.001, odds ratio = 3.92, 95%CI[2.18, 9.85]).



Chapter 3. Tactile stimulation of individuals in an ant colony

Figure 3.2: The first row of plots illustrates the probabilities of ants to respond to the dummy in the different density zones (A), to a fresh and an acclimatized dummy (B) and while doing different tasks (C). Horizontal blue lines represent fitted values calculated by the model and vertical lines represent 95% confidence intervals calculated via bootstrapping. Stars (*, ** or ***) denote significant differences between groups and a p-value lower than 0.05, 0.1 and 0.001, respectively. The second row of plots illustrates the probability of ants to respond to the dummy as a function of foraging frequency (D), brood care frequency (E) and guarding time (F) z-transformed. Each dot corresponds to an individual ant and the dots' colors represent colony identity. The dashed line is the fitted model and the grey zone the 95% confidence intervals of the model calculated via bootstrapping.

3.3.2 Factors that influence the probability of alarm behavior

Next, we tested whether the probability of responding alarmed towards the moving dummy was influenced by the density of workers, acclimatization of the dummy, task performed just before the interaction, brood care frequency, foraging frequency and guarding time. The comparison of the full and null models revealed that the predictors were significant as a whole ($\chi^2 = 51.24$, df = 10, p < 0.001, Figure 3.3). Regarding the individual fixed effects, we found that the ant density in the zone where the interaction took place had a significant effect on the probability that ants responded alarmed towards the dummy (Likelihood ratio test = 34.59, df = 2, p < 0.001). The probability of ants to respond alarmed towards the dummy was not significantly affected by the task the ants were performing at the time of the interaction (Likelihood ratio test = 5.56, df = 4, p = 0.23) nor by the acclimatization of the dummy (Likelihood ratio test = 3.47, df = 1, p = 0.06). Instead, the probability of ants to respond alarmed towards the dummy dister to test = 3.9, df = 1, p = 0.047) and foraging frequencies (Likelihood ratio test = 5.31, df = 1, p = 0.02). Guarding time did not significantly affect ants' probability to respond alarmed (Likelihood ratio test = 2.53, df = 1, p = 0.11).

Ants were significantly more likely to be alarmed when the interaction took place in a middle-(odds ratio = 0.19, p < 0.001) or low-density zone (odds ratio = 0.17, p < 0.001) than in a high-density zone. There were no differences between middle and low-density zones (odds ratio = 0.91, p = 0.93). As for Model 3, we conducted pairwise tests to exclude the possibility that these differences in alarm may stem from intrinsic differences in response thresholds between ants located in the different parts of the nest. To this end, we conducted follow up tests only considering ants that had been tested both in a high- and middle-density zone or in a high- and a low-density zone. Pairwise tests confirmed the previous results when all observations per ant and zones were considered as well as when a single observation per ant and zone was considered (Table 3.5).

Density zone	Sample per ant/task	Nants	McNemar- χ^2	Odds ratio	95%CI	р
High-Mid	all observations	53	51.25	0.09	0.01, 0.17	<0.001
High-Low	all observations	53	54.79	0.12	0.04, 0.22	<0.001
High-Mid	one observation one observation	53	30.62	0.05	-0.29, 0.15	<0.001
High-Low		53	19.12	0.17	0.03, 0.36	<0.001

Chapter 3. Tactile stimulation of individuals in an ant colony

Table 3.5: Pairwise comparisons of the probability to respond alarmed of ants tested in two density zones.



Figure 3.3: The first row of plots illustrates the probabilities of ants to respond alarmed towards the dummy in the different density zones (A), to a fresh and an acclimatized dummy (B) and when doing different tasks (C). Horizontal blue lines represent fitted values calculated by the model and vertical lines represent 95% confidence intervals calculated via bootstrapping. Stars (***) denote significant differences between groups and a p-value lower than 0.001. The second row of plots illustrates the probability of ants to respond alarmed towards the dummy as a function of foraging frequency (D), brood care frequency (E) and guarding time (F) z-transformed. Each dot corresponds to an individual ant and the dots' colors represent colony identity. The dashed line is the fitted model and the grey zone the 95% confidence intervals of the model calculated via bootstrapping.

3.3.3 Habituation hypothesis

The observed differences between density zones in the probability of responding to the dummy and the probability of responding alarmed could stem from a habituation process, whereby ants in higher density zones are exposed to more interactions and thus become less responsive (including less alarmed) towards stimuli. To test this hypothesis, we evaluated the effect that the number of previous interactions had on the probabilities of responding and being alarmed. We fit a series of binomial models including as fixed effect predictors the number of interactions that each ant experienced in three different time windows (one, two and five minutes before the interaction). We controlled for colony identity and included the random intercept of individual identity. The number of interactions that an ant had during the minute before had a negative effect on the probability of responding to the dummy (estimate = -0.45, $\chi^2 = 16.85$, df = 1, p < 0.001). This effect was still significant when the time window considered was expanded to two minutes (estimate = -0.45, $\chi^2 = 14.67$, df = 1, p < 0.001) and five minutes before the ant-dummy interaction (estimate = -0.067, $\chi^2 = 7.96$, df = 1, p = 0.004), although to a lesser extent. The number of interactions that an ant had during the minute before an interaction also had a negative effect on the probability of responding alarmed towards the dummy (estimate = -0.32, $\chi^2 = 7.94$, df = 1, p = 0.005). However, this effect was not significant when the time window considered was extended to two minutes (estimate = -0.19, $\chi^2 = 2.92$, df = 1, p = 0.09) and five minutes (estimate = -0.17, χ^2 = 2.33, df = 1, p = 0.13).

To rule out the possibility that these differences may stem from intrinsic behavioral differences associated with the overall level of activity of different ants, we investigated whether for a given ant, the probabilities of response and alarm were influenced by the ant's activity level. We classified interactions as having taken place in a high- or low-activity window if the number of interactions of the ant was higher or lower (respectively) compared to its average activity level measured during the 12 hours preceding the interactions and the 12 hours after the interaction. These analyses showed that ants were less likely to respond to the dummy when the interaction took place during a high-activity than low-activity window, regardless of window length (Table 3.6). The ants' probability of responding alarmed towards the dummy was also lower when

Window [min]	Dependent variable	McNemar- χ^2	Odds ratio	95%CI	р
1	Prob. response	93.94	5.69	[3.94, 9.34]	< 0.001
2	Prob. response	92.92	4.67	[3.49, 7.47]	< 0.001
5	Prob. response	117.04	6.71	[4.66, 11.12]	< 0.001
1	Prob. alarm	84.85	0.24	[0.16, 0.33]	< 0.001
2	Prob. alarm	83.43	0.26	[0.18, 0.36]	< 0.001
5	Prob. alarm	64.57	0.28	[0.19, 0.39]	< 0.001

Chapter 3. Tactile stimulation of individuals in an ant colony

Table 3.6: Comparisons of ants' probabilities of responding to the dummy (Prob. response) and of responding alarmed towards the dummy (Prob. alarm) during high and low activity time windows before interaction.

interactions took place in low-activity, regardless of the window length considered (Table 3.6).

3.4 Discussion

Using the Antbot system, we investigated how task performance, social environment and stimuli characteristics influence behavioral responses in the ant *L. acervorum*. Ants were more likely to interact, but less likely to explore, moving than static dummies. This finding is likely to reflect a larger variation in the behavioral responses of ants when the dummy was moving compared to when it was stationary. In other words, even though the ants responded less to the non-mobile dummy compared to the moving dummy, when they did respond to it, they mostly explored it via antennation whereas they responded in a variety of ways to the moving dummy. Similar patterns have been found when comparing the mean tail beat frequency (a measure of responsiveness) of Atlantic salmon (*Salmon salar*) to a moving and a passively floating robot (Kruusmaa et al., 2020). Salmon were more responsive (i.e. had a higher tail beat frequency) towards a moving robot compared to a floating robot. Different responses towards moving and non-moving robotic dummies were also found in electric fish (*Mormyrus rume*) which were more likely to swim towards moving than static dummies (Worm et al., 2017).

When considering only targeted interactions between moving dummies and ants, we found that ants' probability to respond to the dummy varied depending on the task they were performing, with ants doing brood care being the least likely to respond. In comparison, ants that were inactive (idle or frozen, (Charbonneau et al., 2017)) at the time of interaction had a higher probability of responding to the dummy. These results could be explained by the existence of different response thresholds in different specialized groups of ants (inactive vs broodcaring ants, (Charbonneau et al., 2017)). However, individual-level analysis of ants tested when performing different pairs of tasks revealed that the different response probabilities observed were not an individual trait but rather plastically modified depending on the task being performed. Similarly, previous studies have shown that in the context of dyadic behavioral essays with individuals from different colonies, the probability of aggression of Argentine ants (*Linepithema humile*) significantly decreased if brood was present (Buczkowski and Silverman, 2005). The authors report that when brood was present, workers often prioritized carrying brood and tending to the brood over attacking each other. Therefore, ants in our experiments might have prioritized continue tending to the brood or moving the brood away from the dummy over responding to it which would lead to lower response probabilities compared to ants that were mostly inactive at the time of testing.

Ants' probability of response was also influenced by the social environment, namely the density of ants in the zone where the interaction took place. When ants were targeted in high density zones, they were less likely to respond than when they were targeted in middleor low-density zones. The same was true when the probability of ants to respond alarmed towards the dummy was investigated: ants were less likely to respond alarmed when tested in high- compared to middle- or low-density zones. Moreover, there was a negative relationship between the number of interactions an ant had been involved in before being targeted by the dummy and the probability of response and alarm. This relation could be due to two mechanisms. First, more active ants that engaged in a higher number of interactions may have higher response and alarm thresholds compared to less active ants that engaged in a lower number of interactions. Alternatively, there may be a habituation process at play where ants' response probabilities are negatively affected by the previous number of interactions. To disentangle between these two explanations, we compared individual responses during above-average and below-average periods of activity and found that response and alarm probabilities were in fact lower during above-average periods of activity. Taken together these results suggest that desensitization or habituation, a simple form of associative learning that involves a reduction in responsiveness to a repeatedly perceived stimulus (Grangier et al., 2007), influences workers responses to tactile stimuli.

Habituation processes to odorant stimuli have been previously described in ants. In *Pheidole* desert ants the mean level of aggression between ants of different colonies was significantly lower if the ants had interacted before (Langen et al., 2000). Similarly, in the Argentine ant repeated exposure to their alarm pheromone reduced alarmed responses until after four or five exposures, there was no behavioral response at all (Maccaro et al., 2020). Tactile habituation to non-harmful stimuli has been reported in plants (*Mimosa pudica*, (Serpell and Chaves-Campos, 2022)) humans (Kisilevsky and Muir, 1984) and animals (Carew et al., 1972). Although tactile habituation has rarely been investigated in insects, (Hölldobler et al., 1990) proposed that the variation in antennal postures exhibited by ants during trophallaxis could be explain by a necessity to avoid habituation to the tactile stimuli. Our results contrast with previous studies on ant aggression showing that experimentally increased worker density in the ant *Crematogaster mimosae* led to higher levels of aggression towards heterospecifics (*C. nigricepts*, (Ruiz-Guajardo et al., 2017)). However, these differences could be due to methodological differences such as the use of a dummy as a stimulus as well the location of the interactions (i.e., inside vs outside the nest).

The probability of alarm, but not the probability of response, was found to be positively related to task specialization: ants that engaged in more brood care and, to a lesser degree, more foraging were more likely to respond alarmed to the dummy. These results contrast with our findings regarding task effects in the sense that while we found that the probability to respond to the dummy was context-specific, the probability of alarm seems to vary at the individual level. *L. acervorum* has been recently described as having a proportion of specialized individuals of around 15% based on their spatial use patterns (Richardson et al., 2022), indicating that most individuals are facultative, non-specialized workers. Our results suggest that task specialization is related with higher probabilities of alarm in this species.

(Richardson et al., 2022) described the nest entrance as being a key location for *L. acervorum* in the sense that together with the brood pile, this area is visited regularly by the same individuals. The role of immobile individuals at the nest entrance is unclear: these individuals could be foragers waiting to be recruited, individuals tasked with nest defense or intermediates in the distribution of food and information towards nest-workers. Using the dummy as tactile stimuli, we did not find that ants that spent more time guarding were more likely to respond alarmed, a relationship that we did find to be significant when we considered foraging frequency. If guarding and foraging were performed by the same individuals we would have expected these two variables to be equally related to the probability of alarm. Therefore, our results do not support the hypothesis that ants at the nest entrance are foragers. Similarly, we do not find support for the hypothesis that individuals at the nest entrance engage in nest defense, as guarding time does not significantly predict probability of alarm. Instead, these individuals could be acting as intermediates between foragers and intra-nest workers although this hypothesis remains untested.

Ants rely on cuticular chemical profiles to discriminate between nestmates and outsiders. Recognition of outsiders is often followed by alarmed and aggressive responses (Lenoir et al., 2001; Sturgis and Gordon, 2012). Thus, it was surprising that the acclimatization of the dummy did not affect the probability of response nor alarm in our experiments and that ants were equally likely to respond to a freshly made dummy than to a dummy that had spent a week inside of the colony. It is possible that *L. acervorum* ants did not perceive a difference between the chemical profiles of fresh and acclimatized dummies. In the ant *Camponotus herculeanus*, it has been shown that the addition, but not the subtraction, of cuticular carbohydrates elicits aggressive responses. This indicates that at least in this species, ants do not specifically react to nestmates' profiles but instead recognize and reject individuals carrying profiles with elements that do not belong to their colony's profile (Guerrieri et al., 2009). If this was also the case in *L. acervorum*, ants would not have recognized the fresh dummy as foreign because it did not carry novel hydrocarbons but instead had no hydrocarbon profile. Another possible explanation is that the dummies that spent a week inside of the nest prior to the experiments did not acquire the colony's chemical profile, and thus were indistinguishable from the freshly made dummies.

3.4.1 Conclusion

The Antbot platform allows to apply controlled tactile stimulus to targeted ants in order to simultaneously investigate multiple drivers of individual behavioral variation in an ecologicallyrelevant context. Using this platform, we found that, while *L. acervorum's* responsiveness is mediated by context-specific factors such as the task being performed when a stimulus is applied, ants probability of alarm is influenced by individual task specialization. Furthermore, we found evidence that ants might be subjected to a process of tactile habituation where the more interactions an ant has experienced in the minutes preceding the presentation of a stimuli, the lower is the probability that the ant will respond to the stimulus. This result highlights the importance of considering the social environment of individual ants when evaluating response and alarm thresholds.

This study showcases the advantages of presenting tactile stimuli to ants via a magneticallyactuated dummy over using other types of agents such as live or dead conspecifics. The use of a teleoperated dummy allows researchers to target specific individuals in specific areas of the nest instead of performing behavioral essays in separate arenas outside of the ants' social and spatial context. The dummy also allows to provide standardized stimuli to all targeted individuals, an aspect that is hard to replicate when using live or dead agents as these might change over time across trials in terms of, for instance, behavior or odour. The fact that we were able to capture individual behavioral variation and establish causal links between behavior and social- and context-specific factors validates the usefulness of this system to investigate ant behavior in the laboratory. Furthermore, the experiments presented here open new research avenues to investigate the effects of more complex stimulus, such as the combination of tactile and chemical signals, on ant behavior.

4 Socially aware robot navigation

The work described in this chapter represents a standalone project on the abstract problem of robot navigation and will be submitted as:

Temporally persistent motion planning in dense dynamic environments for socially aware navigation Matthias Rüegg¹, Ömür Arslan², Mahmut Selman Sakar¹

¹ Institute of Mechanical Engineering, École polytechnique fédérale de Lausanne, Lausanne, 1015, Switzerland.

² Department of Mechanical Engineering, Eindhoven University of Technology, P.O. Box 530, 5600 MB, Eindhoven, The Netherlands.

Abstract

Temporally consistent motion planning is crucial for navigating robots in dynamic social environments without disturbing the surrounding living or robotic agents. In this chapter, we propose a new approach for achieving temporal persistence in navigation by enforcing structural persistence in high-level motion planning based on simplicial discrete abstractions. We introduce a novel notion of simplicial sphere world complexes for modelling dynamic environments consisting of convex polygonal static obstacles and spherical moving obstacles. Simplicial abstractions can be built with the desired global properties. Here, we use a regular triangulation that takes the vertex weight into account to establish a minimal clearance for the transition of adjacent cells. Accordingly, we describe algorithms for constructing high-level navigation paths and corridors over a simplicial sphere world complex, and show how to realize such high-level plans using low-level feedback control policies for fully-actuated and unicycle robot models. We demonstrate the effectiveness of the algorithms with numerical simulations.

4.1 Introduction

Navigation of mobile robots in crowded public spaces has gained attention due to the increased demand to integrate robots into the social life. In addition to safety, energy conservation and efficiency, a fourth consideration for navigating such environments is social awareness, which ensures that the robot is navigated in a socially compliant manner (Kruse et al., 2013). There is no consensus on the precise definitions of the factors that are necessary for socially aware navigation (Rios-Martinez et al., 2015). Yet, a working definition of the problem together with the fundamental metrics to evaluate the performance of a social robot navigation system are formulated (Mavrogiannis et al., 2021). These metrics include path efficiency (i.e., the ratio of the costs of sub-optimal and optimal paths), collision rate, speed efficiency (i.e., the ratio of nominal and actual speeds) and path irregularity (i.e, the amount of deviation from a straight path). Based on these metrics, the primary objective of socially aware navigation is to reduce unnecessary interactions between the robot and the surrounding agents while providing consistent motion to avoid erratic and oscillatory motion due to uncoordinated mutual avoidance (Feurtey, 2000) or freezing due to excessive conservatism caused by uncertainty in prediction (Trautman et al., 2015). Predictable motion is expected to increase the probability of acceptance of the robot into the crowd (Lichtenthäler et al., 2012).

Navigation has been studied as a hierarchical planning problem that consists of global path planning and local motion planning. Given a route found by a global path planner, a local motion planner aims to safely navigate to a next waypoint while avoiding mobile agents and satisfying various dynamic constraints. Socially aware navigation algorithms can be grouped according to the degree of coupling between prediction and planning (Kruse et al., 2013; Mavrogiannis et al., 2021). Recent research has shown that coupling robot action with predictions on the movement of the agents can avoid erratic motion and the freezing problem (Kruse et al., 2013; Cai et al., 2020; Mavrogiannis et al., 2021). A representative example is the combined prediction-actuation approach used with *velocity obstacles* (Fiorini and Shiller, 1998; Van den Berg et al., 2008; Wilkie et al., 2009; Gonon et al., 2021). These reactive navigation strategies are commonly tested in relatively simple environments that include small number of agents or with robots moving in well-structured environments such as along a corridor (Huber et al., 2022; Feder and Slotine, 1997; Fiorini and Shiller, 1998; Phillips and Likhachev, 2011; Gonon et al., 2021).

Path planning for mobile robots in large dynamic environments is a challenging problem, as the robots are required to efficiently reach their given goals while simultaneously avoiding potential conflicts with other robots or dynamic objects. Traditional solutions usually treat dynamic agents as static obstacles, and re-call a planning algorithm to search for an alternative path whenever the robot encounters a conflict. However, such re-planning strategies often cause unnecessary detours and sacrifice optimality (Phillips and Likhachev, 2011). To address this problem, approximate path planning strategies based on reinforcement learning (Wang et al., 2020), model predictive control (Mavrogiannis et al., 2022), probabilistic roadmaps (Foka and Trahanias, 2010), rapidly exploring random trees (Fulgenzi et al., 2008) and neural networks (Villacorta-Atienza and Makarov, 2013) have been proposed.

Combinatorial algorithms guarantee to rapidly find optimal solutions if they exist, and report if they do not exist ((LaValle, 2006)). Discrete abstractions can transform continuous motion planning into a computationally inexpensive problem without losing any essential information. For a finite size agent, if the assumption that the workspace is fully permissible is dropped, poor abstraction will lead to problems in path planning. The basic idea is that a *polygonal sphere world* can be discretized using triangulation in a way that it encodes the penetrability, and its dual graph can then be used for path planning. Meshes have been used to encode the distance to collision for navigation in dynamic environments with polygonal obstacles (Van Toll et al., 2012). Discrete abstractions with temporal logic specifications have been developed for motion planning in complex environments with sequential temporal task specifications and collision avoidance (Burridge et al., 1999; Belta et al., 2005; Fainekos et al., 2005a,b; Kress-Gazit et al., 2009; Conner et al., 2009).

In this chapter, we introduce a framework in which solutions of discrete algorithms dealing with the complexity of the environment automatically ensure socially aware robot navigation. We assume complete and instantaneous knowledge on the world, and focus on the synthesis of a global path planner suited for a dense dynamic world. The main idea is to maximize path efficiency through temporal persistence. We also introduce the clearance penalty concept, which avoids freezing due to the temporary non-existence of a collision free path.

The abstraction of the polygonal sphere world is based on a meshing concept that implicitly encodes the collision distance in order to estimate the navigation cost. The concept of the *generalized sphere world*, a disk shaped workspace punctuated by spherical obstacles, was originally introduced by (Rimon and Koditschek, 1992) together with potential functions for exact navigation of a point robot. Here, we use a similar concept, but combine spherical moving obstacles with static polygonal obstacles for a more holistic representation of the workspace. The algorithm traces and re-evaluates previous path solutions efficiently on the structural level of a simplicial abstraction, and uses a heuristic to improve path efficiency. The path planning algorithm is compatible with any motion planner that is capable of pursuing a local reference goal in a polygonal sphere world. In this work, we integrated a motion controller that locally modifies the cell decomposition in order to establish a collision free region, consequently providing a safe and provably exact control policy (Arslan and Koditschek, 2016).

The main contributions of the chapter are summarized below:

1. Modeling of navigation world with a simplicial discrete abstraction of the convex workspace including spherical and convex polygonal obstacles and use its duality for

78

path planing on the dual graph to find what we call navigation corridors.

- 2. Construction of a finite-size simplicial spherical world using regular triangulations (RT) to model the connectivity and estimate the minimal clearance of simplicial corridors.
- 3. Achieving temporal persistence in motion planning via a cost heuristic on retained previous (at timestep $t \Delta t$) and current optimal (at timestep t) solutions.
- 4. Implementation of planning with the path following of a collision safety enforcing motion controller for fully-actuated and unicycle robot dynamics.
- 5. Simulations to demonstrate the algorithm in multi-robot navigation and cluttered dynamic worlds.



Figure 4.1: A schematic overview of the temporally persistent motion planning algorithm.

The rest of the chapter is organized as follows. In Section 4.2, the simplicial sphere world is introduced along with the basic definitions for the conforming constrained triangulation that is used as an abstraction. A certificate for collision free corridor in sphere worlds is derived in

Section 4.3 as a basis for the cost function used to find and re-identify navigation path on the motion graph in Section 4.4. In Section 4.5, the capabilities of the algorithm are explored using a multi-agent and a single agent scenario in a complex dynamic world. Section 4.6 concludes the chapter.

4.2 Simplicial sphere world complexes

Figure 4.2 illustrates the inspiration from designing a navigation algorithm for an autonomous agent in the ant nest for the abstract problem addressed in this Chapter. The goal for the robotic dummy agent (black/green circle) is to contact a given target individual (red circle) autonomously. Ants live densely packed inside a nest, together with a queen and brood and move in way which is hard to predict or interpret. The forbidden areas, e.g. to protect brood and queen, are represented with polygonal regions. The algorithm presented here assumes convex polygons, which can however be used in a union to model non-convex shapes.



Figure 4.2: The challenge of finding a navigation path in the ant world, modeled with the polygonal sphere world. Ant are represented with moving spheres, workspace and forbidden areas, e.g. to protect the queen (gold colored individual) and the brood (grey points), are represented with polygonal regions.

4.2.1 Definitions and notation

Definition 1 (Sphere) A sphere centered at x with radius $r \in \mathbb{R}_{\geq 0}$ is defined as:

$$\mathbf{s}(\mathbf{x}, r) := \{ \mathbf{q} \in \mathbb{R}^n \mid \|\mathbf{q} - \mathbf{x}\| = r \}$$
(4.1)

and represents the surface of the closed ball $B(s) := \{q \in \mathbb{R}^n \mid ||q - x|| \le r\}.$

Definition 2 (Orthogonal spheres) Two spheres s_i , s_j are said to be (*sub-*) orthogonal, if

$$\|\mathbf{x}_{i} - \mathbf{x}_{j}\|^{2} \ge = r_{i}^{2} + r_{j}^{2}$$
(4.2)

Definition 3 (Power distance (Edelsbrunner and Shah, 1996)) The power distance of a point $p \in \mathbb{R}^n$ to the sphere s, centered at $x \in \mathbb{R}^n$, with radius $r \ge 0$, is defined as:

$$\pi(s, p) = \|\mathbf{x} - p\|^2 - r^2$$
(4.3)

The power distance can be interpreted as the length of a segment from point p to a tangent point on s. Definitions 2 and 3 imply that the centers of two orthogonal spheres s_i, s_j lie at equal power distance from each other: $\pi(s_i, x_j) = \pi(s_j, x_i)$.

Definition 4 (Triangulation) A triangulation¹ $\mathcal{T}(\mathbf{x}) = \{t_1, \dots, t_k\}$ of a finite set of m > 0 points $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_m) \in \mathbb{R}^{m \times n}$ in a *n*-dimensional Euclidean space is a collection of vertex indices $t_1, \dots, t_k \in P_n(m) := \{t \subseteq \{1, \dots, m\} \mid |t| = n + 1\}$ of *n*-simplices $\Delta_{t_1}, \dots, \Delta_{t_k}$ such that

- $\bigcup_{i=1}^{k} \Delta_{\mathbf{t}_i}(\mathbf{x}) = \operatorname{conv}(\mathbf{x}_1, \dots, \mathbf{x}_m)$
- $\bigcup_{i=1}^{k} \mathbf{t}_{i} = \{1, ..., m\}$
- $\mathring{\Delta}_{t_i}(\mathbf{x}) \cap \mathring{\Delta}_{t_i}(\mathbf{x}) = \emptyset, \quad \forall i \neq j \in \{1, \dots, k\}$

where the *n*-simplex $\Delta_t(\mathbf{x})$ is defined as

$$\Delta_{\mathsf{t}}(\mathbf{x}) := \operatorname{conv}(\{\mathbf{x}_i | i \in \mathsf{t}\}) \tag{4.4}$$

and $\mathring{\Delta}_{t_i}(\mathbf{x})$ denotes the interior. For an edge the expression $[x_j, x_k] := \operatorname{conv}(\{x_i | i \in \{j, k\}\})$ is used.

Definition 5 (Constrained Triangulation) Given a set of constraints $\mathscr{C} \subset \bigcup_{i=0}^{n} P_{i}(m)^{2}$, a *constrained* triangulation is a triangulation $\mathscr{T}(\mathbf{x}) = \{t_{1}, \dots, t_{m}\}$, such that for every $c \in \mathscr{C}$ there exists a triangle $t \in \mathscr{T}$ that satisfies $c \subseteq t$. \mathscr{T} is said to *respect* \mathscr{C} .

Definition 6 (Conforming Constrained Triangulation) Given a constrained triangulation $\mathcal{T}(\mathbf{x})$ that respects a set of constraints \mathscr{C} , a *conforming constrained triangulation*³ $\mathcal{T}'(\mathbf{x} \cup \mathbf{x}')$ is a refined version of the triangulation \mathcal{T} with an set of m' additional vertices⁴ \mathbf{x}' such that every constraint is a union of edges in \mathcal{T}' . More precisely, for each $c_i \in \mathscr{C}$ there exists a subset $E = \{P_{n-1}(m+m')\} \subset \{t\} \subset \mathcal{T}'$, such that $\operatorname{conv}_E(\mathbf{x} \cup \mathbf{x}') = \operatorname{conv}_{c_i}(\mathbf{x})$. \mathcal{T} is said to *conform* to \mathscr{C} .

Note that for an appropriate a priori choice of \mathbf{x}' and $\mathscr{C}'(\mathbf{x} \cup \mathbf{x}')$ with $\bigcup_{c_i \in \mathscr{C}'} \operatorname{conv}_{c_i}(\mathbf{x} \cup \mathbf{x}') = \bigcup_{c_i \in \mathscr{C}} \operatorname{conv}_{c_i}(\mathbf{x})$, \mathscr{T}' is simply a constrained triangulation of the point $\mathbf{x} \cup \mathbf{x}'$ respecting the constraints \mathscr{C}' .

¹A triangulation in a *n*-dimensional space defines a homogeneous simplicial *n*-complex.

²Formally, in the field of computational geometry, the complex \mathscr{C} describes a *piecewise linear complex* (PLC): a set of vertices, edges and polygons. The general definition of a PLC is given in Appendix A.3.2 for completeness. ³sometimes called *Steiner triangulation* or simply *mesh* (Cheng et al., 2013)

⁴sometimes called *Steiner points* (Cheng et al., 2013)

Definition 7 (Regularity) Consider a triangulation $\Re_{n+2}(\mathbf{s})$ of $|\mathbf{s}| = n + 2$ spheres $\mathbf{s} = (\mathbf{x}, r)$, which is composed of two simplices \mathbf{t}_a and \mathbf{t}_b with $|\mathbf{t}_{(a,b)}| = n + 1$. Then each simplex possesses exactly one vertex that is not part of the shared face $f_{\mathbf{t}_a \mathbf{t}_b} = \mathbf{t}_a \cap \mathbf{t}_b$. For every sphere set \mathbf{s}' with $|\mathbf{s}'| = n + 1$ there exists a unique sphere $\mathbf{s}_{\perp}(\mathbf{s}')$, called *orthosphere*, that is orthogonal to all spheres $\mathbf{s} \in \mathbf{s}'$. \Re_{n+2} (or $f_{\mathbf{t}_a \mathbf{t}_b}$ equivalently) is called *regular* if:

$$\pi(\mathbf{s}_{\perp}(\mathbf{s}_{t_a}), \mathbf{x}_{t_b \setminus t_a}) \ge r_{t_b \setminus t_a}^2 \text{ and } \pi(\mathbf{s}_{\perp}(\mathbf{s}_{t_b}), \mathbf{x}_{t_a \setminus t_b}) \ge r_{t_a \setminus t_b}^2$$

$$(4.5)$$

In other words, $s_{t_b \setminus t_a}$ is sub-orthogonal to $s_{\perp}(\boldsymbol{s}_{t_a})$ and vice-versa.

Definition 8 (Conforming constrained regular triangulation) Given a triangulation $\mathscr{R}(\mathbf{s})$ that conforms to the constraints \mathscr{C} , \mathscr{R} is a *conforming constrained regular triangulation* if:

$$\mathbf{f}_{\mathbf{t}_i \mathbf{t}_j} = \mathbf{t}_i \cap \mathbf{t}_j \text{ is regular } \forall \ \mathbf{f}_{\mathbf{t}_i \mathbf{t}_j} \in \mathscr{R}$$

$$(4.6)$$

It can be shown that there exists a conforming constrained constrained regular triangulation for any set sub-orthogonal spheres **s** and non-intersecting set of constraints \mathscr{C} (Cheng et al., 2013).

Definition 9 (Dual graph of a triangulation) The simplex adjacency of a triangulation $\mathcal{T} = \{t_1, \dots, t_k\}$ is described by the dual graph $G_{\mathcal{T}} = (V, E)$, with nodes $V = \{t_1, \dots, t_k\}$ and edges:

$$E = \{\{\mathbf{t}_i, \mathbf{t}_j\} \mid i \neq j \in \{1, \dots, k\} \text{ and } |\mathbf{t}_i \cap \mathbf{t}_j| = n\}$$
(4.7)

4.2.2 Polygonal sphere world

The polygonal workspace environment is defined as

$$\mathcal{W}_p = \operatorname{conv}_{\mathbf{p}_b}(\mathbf{x}) \setminus \{\operatorname{conv}_{\mathbf{p}_i}(\mathbf{x})\},\tag{4.8}$$

where $p_{(b,i)} \in P_n(m) := \{t \subseteq \{1, ..., m\}\}$ are ordered sets with the vertices of the convex workspace boundary and a set of non-overlapping convex obstacles. For convenience, we define $p_{ws} = p_b \cup \{p_i\}$. The workspace \mathcal{W}_p is punctuated by a set of spheres $\mathbf{s} = \{s_1, ..., s_m\}$, centered at \mathbf{x} , with radii $\mathbf{r} \ge 0$.

Definition 10 (Polygonal sphere world) The tuple $\mathscr{S} = (\mathscr{W}_p, \mathbf{s})$ is called *polygonal sphere world*. **Assumption 1** The sphere centers are contained inside the workspace \mathscr{W}_p , i.e $\mathbf{x}_i \in \mathscr{W}_p$, $\forall \mathbf{x}_i \in \mathbf{x}$. **Assumption 2** The spheres in \mathbf{s} are sub-orthogonal to each other:

$$\|\mathbf{x}_{i} - \mathbf{x}_{j}\|^{2} \ge r_{i}^{2} + r_{j}^{2}, \ i \ne j \in \{1, \dots, m\}$$

$$(4.9)$$

This is motivated by the fact that if Assumption 2 holds, there exists a regular triangulation of the sphere world, which is instrumental for the simplicial abstraction of the sphere world introduced below in Section 4.2.3. Later, in Section 4.4 we assume that spheres are mobile and their center coordinates are known at any time.

4.2.3 Simplicial sphere world

For navigation planning, the polygonal sphere world \mathscr{S} is discretized with a triangulation.⁵

Definition 11 (Simplicial sphere world) Given a constrained triangulation $\mathcal{T}(\mathbf{x})$ respecting the constraints p_{ws} of $\mathcal{S} = (\mathcal{W}_p, \mathbf{s})$, the simplices in the inaccessible space are removed to form the *simplicial sphere world*: $\mathcal{T}_{sw} = \{\mathbf{t} \in \mathcal{T} \mid \Delta_{\mathbf{t}} \in \mathcal{W}_p\}.$

⁵A simplex is the generalized notion of a triangle in an arbitrary dimension *n*. Certain results in this chapter are not restricted to n = 2, however we use the term *simplex* interchangeably with *triangle* where appropriate.

Figures 4.3, left and 4.8, left show examples of simplicial sphere world scenarios. In general the triangulation \mathcal{T} of pointset **x**, with or without constraints, is not unique. With the appropriate choice of a condition on the simplices with respect to **x**, \mathcal{T} can be uniquely specified (Cheng et al., 2013). A *Delaunay triangulation* for example, is triangulation in which the circumcircles of all triangles are free of other vertices. With the regularity condition (Definition 7), the Delaunay case is generalized to weighted points or spheres.⁶ This way a notion of the size of the spheres can be encoded in the triangulation, which will allow to establish a condition on the clearance for navigation corridors in polygonal sphere worlds in Section 4.3. Note that in the limit case $r_i \rightarrow 0$, orthogonality (Definition 2) together with the regularity condition (Definition 7) represent the free circumsphere criterion. Definition 8 is therefore a generalization of the Delaunay triangulation for weighted points or spheres in particular.

As it is the case for the circumcenter, the orthocenter $x_{\perp}(\mathbf{s}_{t_i})$ it is not necessarily contained inside the triangle and as a consequence, not all orthocenters x_{\perp} of the triangles in \mathscr{R} are necessarily contained in $\cup_{t \in \mathscr{R}} \triangle_t$, the underlying subspace of \mathscr{R} (Figure 4.3, left). To establish a clearance estimate for navigation in a simplicial sphere world (Section 4.3), the following requirement is useful:

Assumption 3 Given a polygonal sphere world $\mathscr{S} = (\mathscr{W}_p, \mathbf{s})$, there exists a conforming regular triangulation \mathscr{R} , such that for all $\{\mathbf{t} \in \mathscr{R} \mid \Delta_{\mathbf{t}} \in \mathscr{W}_p\}$ it holds that $\mathbf{x}_{\perp}(\mathbf{s}_t) \in \mathscr{W}_p$.

This can be achieved in refining \mathscr{R} by adding unweighted vertices on constraint segments of the workspace boundary $\partial \mathscr{W}_p$ where appropriate, while maintaining the conditions of constrained regularity of Definition 8. An algorithm to construct a conforming regular triangulation such that Assumption 3 is met, is given in Appendix A.4. The resulting complex is shown in Figure 4.3, right. To realize Assumption 3, it is necessary that Assumption 1 is tightened such that obstacle spheres lie strictly inside \mathscr{W}_p . This is motivated in Appendix A.4 as well.

⁶The weight as the interpretation of r^2 .



Figure 4.3: Constrained (left) and conforming (right) regular triangulation. Cyan: constraint segments of \mathcal{W}_p . Red: orthospheres. Vertex sphere radii are omitted.

4.3 Connectivity of regular simplicial sphere worlds

The sphere size encoded in the regular triangulation, in the form of the power distance (Definition 3) and sub-orthogonality criterion (Definition 7), can be interpreted as clearance between sphere in the Delaunay case $(r_i \rightarrow 0)$, in terms of euclidean distance. For finite size spheres, the distance to collision and the connectivity of the sphere world does not follow directly. Here, we first establish a certificate for the connectivity of adjacent simplices, which is sufficient to describe the transition between simplices for point agents, and then further exploit the properties of the regular triangulation to find an estimate of the clearance for finite size size agents.

Definition 12 (Chordale (Edelsbrunner and Shah, 1996)) For two spheres s_i , s_j , the set of loci x with equal power distance:

$$H_{ij} := \left\{ \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}_i - \mathbf{x}\|^2 - r_i^2 = \|\mathbf{x}_j - \mathbf{x}\|^2 - r_j^2 \right\}$$
(4.10)

is called *chordale* and represents an *n*-dimensional hyperplane.

Instead of the power distance, it is in fact possible to define a spatial partitioning based on the euclidean distance.⁷ However, the borders of a cell of this structure are hyperbolic segments and not chordales, which is more complex to treat in order to establish collision free navigation. From Definition 12 follows that the intersection points of colliding spheres lie on the chordale, where the power distance is equally zero to both spheres. The chordales between a given vertex and all other points of a set **s** represent the boundaries of the powercell, which is a useful concept to construct a motion planner (Section 4.5.1).

Definition 13 (Powercell) The powercell of a sphere s in a set of spheres s ∉ s is defined as:

$$P(\mathbf{s}) = \left\{ \mathbf{q} \in \mathcal{W}_p \mid \pi(\mathbf{s}, \mathbf{q}) \le \pi(\mathbf{s}_i, \mathbf{q}), \mathbf{s}_i \in \mathbf{s} \setminus \mathbf{s} \right\}$$
(4.11)

The separating hyperplanes arising from the equalities in Equation (4.11) are the chordale of s and the spheres in **s**. If the spheres of some set **s** are mutually sub-orthogonal, it can be shown that the chordale of every pair of $s_i, s_j \in \mathbf{s}$ separates the centers.⁸ It follows that no power cell is empty, that is $x_i \in P(s_i) \forall s_i \in \mathbf{s}$ (Aurenhammer, 1987).

The power cell therefore captures the collision of a generator sphere (Arslan and Koditschek, 2016). This is sufficient to certify navigation corridors as collision free for point agents. In the following we develop a tight lower bound on the clearance for navigation across a sequence of triangles, which can be used to certify collision free corridors for finite size agents.

First, note that if Assumption 2 (sub-orthogonal spheres) is valid, then the interior of a simplex Δ_t can geometrically only be intersected by spheres centered in the region $\mathbf{x} \in \bigcup_{i \in t} \operatorname{cone}_{i,t}(\mathbf{x})$, where:

$$\operatorname{cone}_{i,t}(\mathbf{x}) = \left\{ \mathbf{x}_i + \theta_1(\mathbf{x}_j - \mathbf{x}_i) + \theta_2(\mathbf{x}_k - \mathbf{x}_i) \mid \theta_i \in \mathbb{R}_{\ge 0} \right\}$$
(4.12)

is the conic hull with apex at x_i and boundaries through (x_i, x_j) and (x_i, x_k) respectively. Below

⁷This construct is called *additively weighted Voronoi diagram* or *Apollonius diagram*.

⁸This condition can in fact be relaxed to mutually excluded sphere centers.

it will be shown that this region can be reduced to $\mathbf{x} \in \bigcup_{i \in \mathbf{t}} \overline{\Delta}_{i,t}(\mathbf{x})$, where:

$$\bar{\Delta}_{i,t}(\mathbf{x}) = \operatorname{cone}_{i,t}(\mathbf{x}) \setminus \Delta_{t}(\mathbf{x}) \tag{4.13}$$

is called *capped cone* of simplex t and apex $i \in t$.

Proposition 1 (*Clearance over cone, tight lower bound*) Given a conforming regular triangulation \mathscr{R} and triangle $t = \{i, j, k\}, t \in \mathscr{R}$, the clearance between the apex sphere s_i and a sphere $s_e = (x_e, r_e)$ with $x_e \in \text{cone}_{i,t}(\mathbf{x})$, is lower bound by:

$$d_{c}(\mathbf{t},i) = \begin{cases} d(\mathbf{s}_{i},[\mathbf{x}_{j},\mathbf{x}_{k}]), & if\{j,k\} \in \mathbf{p}_{ws} \\ \min \begin{cases} d(\mathbf{s}_{i},[\mathbf{x}_{j},\mathbf{x}_{\perp}]), \\ d(\mathbf{s}_{i},[\mathbf{x}_{k},\mathbf{x}_{\perp}]) \end{cases}, & if\mathbf{x}_{\perp} \in \bar{\Delta}_{i,\mathbf{t}} \\ d(\mathbf{s}_{i},\hat{\mathbf{s}}_{\perp}(\mathbf{t},i)), & otherwise \end{cases}$$
(4.14)

Where $d(A, B) = \inf_{x \in A, y \in B} ||x - y||$ is the shortest distance between sets A and B, p_{ws} are the boundary faces of the polygonal workspace, x_{\perp} is the orthocenter of t, and $\hat{s}_{\perp}(t, i) = s_{\perp}(\mathbf{s}_{t \setminus i}, s_{\perp}(\mathbf{s}_{t}))$ is the advanced orthosphere of t for apex *i*, *i.e.* the unique sphere that is orthogonal to the vertices s_{i} , s_{k} and the orthosphere s_{\perp} .

The first condition of Equation (4.14) represents the case where $\operatorname{cone}_{i,t}(\mathbf{x})$ is bounded by the workspace. For latter cases in Equation (4.14), the biggest set $\Gamma \subset \operatorname{cone}_{i,t}(\mathbf{x})$ is identified, such that $\Gamma \cap \Omega = \emptyset$ with $\Omega = \bigcup B(\mathbf{s}_e)$, the union of all potentially existing external spheres \mathbf{s}_e with $\mathbf{x}_e \in \operatorname{cone}_{i,t}(\mathbf{x})$ that respect the regularity of t and sub-orthogonality to $\mathbf{s}_i, \forall i \in t$. The distinction of the second and third case (Figure 4.4) is motivated by an observation that follows from Lemma 1 and 2, derived in the following together with other statements which are instrumental for the proof of Proposition 1. **Lemma 1** In an n-dimensional Euclidean space \mathbb{R}^n , consider $m \le n+1$ sub-orthogonal spheres s_1, \ldots, s_m that are orthogonal to some other sphere s_0 , i.e.,

$$\|\mathbf{x}_{i} - \mathbf{x}_{j}\|^{2} \ge r_{i}^{2} + r_{j}^{2}, \quad \forall i \neq j$$
(4.15)

$$\|\mathbf{x}_{i} - \mathbf{x}_{0}\|^{2} = r_{i}^{2} + r_{0}^{2}, \quad \forall i = 1, \dots, m$$
(4.16)

if a sphere s is sub-orthogonal to s_0, \ldots, s_m , i.e.

$$\|\mathbf{x} - \mathbf{x}_i\|^2 \ge r^2 + r_i^2, \quad \forall i = 0, 1, \dots, m$$
(4.17)

then:

$$\mathbf{x} \notin \operatorname{conv}(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_m) \tag{4.18}$$

Proof. In \mathbb{R}^n , there exist at most n + 1 spheres which are orthogonal to a shared sphere, which is their orthogonal center. Hence we require $m \le n + 1$. It is also useful to observe from sub-orthogonality: $||\mathbf{x} - \mathbf{x}_i||^2 \ge r^2 + r_i^2 \ge r_i^2$, because $r^2 \ge 0$. Accordingly, one can conclude that that x has a nonzero distance to any convex combination of $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_m$:

$$\|\mathbf{x} - \sum_{i=0}^{m} \alpha_i \mathbf{x}_i\|^2 = \|\sum_{i=0}^{m} \alpha_i (\mathbf{x} - \mathbf{x}_i)\|^2 = \|\alpha_0 (\mathbf{x} - \mathbf{x}_0) + \sum_{i=1}^{m} \alpha_i (\mathbf{x} - \mathbf{x}_i)\|^2$$
(4.19)

$$=\alpha_0^2 \|\mathbf{x} - \mathbf{x}_0\|^2 + \sum_{i=1}^m 2\alpha_0 \alpha_i (\mathbf{x} - \mathbf{x}_0)^T (\mathbf{x} - \mathbf{x}_i) + \|\sum_{i=1}^m \alpha_i (\mathbf{x} - \mathbf{x}_i)\|^2$$
(4.20)

$$=\alpha_0^2 \|\mathbf{x} - \mathbf{x}_0\|^2 + \|\sum_{i=1}^m \alpha_i (\mathbf{x} - \mathbf{x}_i)\|^2 + \sum_{i=1}^m 2\alpha_0 \alpha_i (\|\mathbf{x} - \mathbf{x}_0\|^2 + \|\mathbf{x} - \mathbf{x}_i\|^2 - \|\mathbf{x}_i - \mathbf{x}_0\|^2)$$
(4.21)

$$> \alpha_0^2 r_0^2 \ge 0$$
 (4.22)

where α_i are convex combination coefficients, i.e. $\alpha_i \in [0, 1]$ and $\sum_{i=0}^{m} \alpha_i = 1$, and the strict inequality follows from:

- $||\mathbf{x} \mathbf{x}_0||^2 > r_0^2$, for a finite radius sphere r > 0
- $\|\mathbf{x} \mathbf{x}_0\|^2 + \|\mathbf{x} \mathbf{x}_i\|^2 \|\mathbf{x}_i \mathbf{x}_0\|^2 > (r_0^2 + r_i^2 r_0^2 r_i^2) = 0$
- $\|\sum_{i=1}^{m} \alpha_i (\mathbf{x} \mathbf{x}_i)\|^2 \ge 0$

In particular: For a simplex \triangle with n + 1 vertex spheres **s** and orthosphere s_{\perp} , any sphere s_e that is sub-orthogonal to s_{\perp} and all $s \in \mathbf{s}$, it is true that $x_e \notin \triangle$.

Lemma 2 Let s_a and s_b be two orthogonal spheres, i.e. $\|\mathbf{x}_a - \mathbf{x}_b\|^2 = r_a^2 + r_b^2$. If some sphere s is sub-orthogonal to both s_a and s_b , i.e. $\|\mathbf{x}_{(a,b)} - \mathbf{x}\|^2 \ge r_{(a,b)}^2 + r^2$, then it satisfies:

$$\|\alpha x_a + (1 - \alpha x_b) - x\|^2 \ge r^2$$
 for $\alpha \in [0, 1]$ (4.23)

Proof. First expand the norm term as:

$$\|\alpha \mathbf{x}_{a} + (1 - \alpha \mathbf{x}_{b}) - \mathbf{x}\|^{2} = \|\alpha (\mathbf{x}_{a} - \mathbf{x}) + (1 - \alpha)(\mathbf{x}_{b} - \mathbf{x})\|^{2}$$
(4.24)

$$=\alpha^{2} \|\mathbf{x}_{a} - \mathbf{x}\|^{2} + (1 - \alpha)^{2} \|\mathbf{x}_{b} - \mathbf{x}\|^{2} + 2\alpha(1 - \alpha)(\mathbf{x}_{a} - \mathbf{x})^{T}(\mathbf{x}_{b} - \mathbf{x})$$
(4.25)

Now observe that:

$$2(\mathbf{x}_{a} - \mathbf{x})^{T}(\mathbf{x}_{b} - \mathbf{x}) = \|\mathbf{x}_{a} - \mathbf{x}\|^{2} + \|\mathbf{x}_{b} - \mathbf{x}\|^{2} - \|\mathbf{x}_{a} - \mathbf{x}_{b}\|^{2}$$
(4.26)

By combining these two equations with $||\mathbf{x}_a - \mathbf{x}_b||^2 = r_a^2 + r_b^2$ one can conclude that:

$$\|\alpha \mathbf{x}_a + (1 - \alpha \mathbf{x}_b) - \mathbf{x}\|^2 =$$
(4.27)

$$\alpha(2-\alpha) \|\mathbf{x}_a - \mathbf{x}\|^2 + (1-\alpha)(1+\alpha) \|\mathbf{x}_b - \mathbf{x}\|^2 - \alpha(1-\alpha)(r_a^2 + r_b^2)$$
(4.28)

$$\stackrel{(1)}{\geq} \alpha(2-\alpha)(r_a^2+w) + (1-\alpha)(1+\alpha)(r_b^2+w) - \alpha(1-\alpha)(r_a^2+r_b^2)$$
(4.29)

$$=\alpha r_a^2 + (1-\alpha)r_b^2 + (1+2\alpha(1-\alpha))r^2 \ge r^2$$
(4.30)

where (1) follows from sub-orthogonality.

In other words: Lemma 2 states that the segment between the centers of two orthogonal spheres s_a , s_b cannot be intersected by any other sphere, sub-orthogonal to s_a , s_b . This is true in particular in the context of a regular simplex \triangle , for segments between vertices and the the orthocenter x_{\perp} of \triangle (Figure 4.4, blue segments).

The following statements, including Remark 1, Proposition 2 and Lemma 3, are reproduced from (Cheng et al., 2013) and adapted to the notation of this chapter. The proofs are given in (Cheng et al., 2013) and omitted here.

Remark 1 A property of power diagrams is that any cross-section of a power diagram is a lower-dimensional power diagram of a modified set of spheres that lie in the cross-section. The modified spheres are found by orthogonal projection of the spheres onto the cross-sectional affine subspace and adjusting their radii as described in the following proposition.

Proposition 2 (modified spheres (Proposition 7.3 in (Cheng et al., 2013))) Let $\Pi \subset \mathbb{R}^n$ be an affine subspace. Let $s_p \in \mathbb{R}^n$ be a sphere centered at x_p with radius r_p . Let x'_p be the orthogonal projection of x_p onto Π , where the radius satisfies $r'^2_p = r^2_p - ||x_p - x'_p||^2$. Then for every point $x \in \Pi$ it holds:

$$\pi(\mathbf{s}_p, \mathbf{x}) = \|\mathbf{x}_p - \mathbf{x}\|^2 - r_p^2 \stackrel{(1)}{=} \|\mathbf{x}_p' - \mathbf{x}\|^2 + \|\mathbf{x}_p - \mathbf{x}_p'\|^2 - r_p^2 = \|\mathbf{x}_p' - \mathbf{x}\|^2 - r_p'^2 = \pi(\mathbf{s}_p', \mathbf{x})$$
(4.31)

where (1) is explained by the orthogonal projection.

For spheres with real radii (i.e. $r^2 \ge 0$), the projected sphere s'_p can be interpreted as the smallest *n*-ball that includes the cross-section $B(s_p) \cap \Pi$. The same principle applies not only to weighted sites, but also to orthospheres. The following proposition shows that cross-sections of an orthosphere of a simplex indicate the diametric orthospheres of its faces. This is illustrated in Figure 4.5 by the dashed segment and circles.

Lemma 3 (Orthosphere Restriction (Lemma 7.4 in (Cheng et al., 2013))) Let t be a simplex with weighted vertices in \mathbb{R}^n . Let s_{\perp} be an orthosphere of t, and let f be a face of t. Let x'_{\perp} be the orthogonal projection of x_{\perp} onto the affine hull of f. Then the orthocenter and orthoradius of f are x'_{\perp} and r'_{\perp} , respectively, where $r'^2_{\perp} = r^2_{\perp} - ||x_{\perp} - x'_{\perp}||^2$, and x'_{\perp} is the diametric orthocenter of f.

On an edge between spheres s_a, s_b , the largest occupied segment $[x_a, x_b] \cap B(s)$ by any sphere s sub-orthogonal to both s_a and s_b , is $[x_a, x_b] \cap B(s_{\perp})$, with s_{\perp} exactly orthogonal to s_a and s_b . To see this, observe that for a point $x' \in [x_a, x_b] \cap B(s)$ to minimize either $||x_{(a,b)} - x'|| \ge$ $\|\mathbf{x}_{(a,b)} - \mathbf{x}\| - r \ge 0$, the distance as well as the radius of s must be maximal. The lower bound is thus realized by \mathbf{s}_{\perp} , called *diametric orthosphere* (Figure 4.5). Since by Lemma 3, the diametric orthosphere is the projection of any higher dimensional orthosphere, violation of the orthogonality on a face has the violation of regularity in any dimension as a consequence, as stated in Remark 1. This completes the prerequisites to proof Proposition 1.

Proof. **Proposition 1** The first case of Equation (4.14) is trivial and no formal proof is carried out. For the latter cases, Lemma 1 restricts the area for locations of x_e as $x_e \in \mathbb{R}^2 \setminus \text{conv}(\{x_i, x_j, x_k, x_{\perp}\})$. As stated in Section 4.2.3, the orthocenter x_{\perp} is not necessarily contained inside the simplex Δ , which motivates the distinction of the two cases. The strategy of the proof is to identify for each case, the region Γ which is guaranteed intersection free with any potentially existing external sphere s_e .

case $\mathbf{x}_{\perp} \in \bar{\Delta}_{t,i}$ (Figure 4.4, right): First, note that by Assumption 3, if $\mathbf{x}_{\perp} \notin \Delta_{t}$, it still holds that $\mathbf{x}_{\perp} \in \mathcal{W}_{p}$ which is necessary for the correctness of the following statements. Lemma 2 states that a segment $[\mathbf{x}_{(i,j,k)}, \mathbf{x}_{\perp}]$, between a vertex and the orthocenter, cannot be intersected. Since $[\mathbf{x}_{(j,k)}, \mathbf{x}_{\perp}] \in \partial \operatorname{conv}(\{\mathbf{x}_{i}, \mathbf{x}_{j}, \mathbf{x}_{k}, \mathbf{x}_{\perp}\})$, it follows for the unoccupied region $\Gamma = \operatorname{conv}(\{\mathbf{x}_{i}, \mathbf{x}_{j}, \mathbf{x}_{k}, \mathbf{x}_{\perp}\}) \neq \operatorname{conv}(\{\mathbf{x}_{i}, \mathbf{x}_{j}, \mathbf{x}_{k}\})$ (Figure 4.4, right, blue patch). The region $\Omega = \bar{\Delta}_{i,t} \setminus \Gamma$ can be occupied by infinite spheres (Figure 4.4, right, grey patch). The estimate for $d_{c}(t, i)$ follows immediately.

case $\mathbf{x}_{\perp} \notin \overline{\Delta}_{t,i}$ (Figures 4.4, left and 4.5):

Here $[\mathbf{x}_{(j,k)}, \mathbf{x}_{\perp}] \notin \partial \operatorname{conv}(\{\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, \mathbf{x}_{\perp}\})$ and the segment $[\mathbf{x}_j, \mathbf{x}_k] \in \partial \operatorname{conv}(\{\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, \mathbf{x}_{\perp}\})$ can be intersected by an external sphere \mathbf{s}_e . From Remark 1 and Proposition 2 follows that the orthogonal projection of an external sphere \mathbf{s}_e on the face $f_{ik} \in t$, must respect sub-orthogonality w.r.t. $\mathbf{s}(f_{jk})$ as well. \mathbf{s}_{\perp} is unique and with Lemma 3 it follows that the intersected segment $[\mathbf{p}_j, \mathbf{p}_k] = [\mathbf{x}_j, \mathbf{x}_k] \cap B(\mathbf{s}_{\perp})$ is uniquely defined as well. Therefore any sphere, and in particular \mathbf{s}_e , must respect $B(\mathbf{s}_e) \cap [\mathbf{x}_j, \mathbf{x}_k] \subseteq [\mathbf{p}_j, \mathbf{p}_k]$ (Figure 4.5). Now assume \mathbf{s}_e is orthogonal to $\mathbf{s}_j, \mathbf{s}_k$, therefore $\mathbf{s}_e \cap [\mathbf{x}_j, \mathbf{x}_k] \equiv [\mathbf{p}_j, \mathbf{p}_k]$ and in addition it is also orthogonal to \mathbf{s}_{\perp} . Further assume that there exists an external sphere \mathbf{s}'_e and a point $\mathbf{p} \in B(\mathbf{s}'_e)$ with $\mathbf{p} \in \Delta_t \setminus B(\mathbf{s}_e)$. A sphere \mathbf{s}'_e



Figure 4.4: Estimate of collision distance $d_c(t, i)$ over $\operatorname{cone}_{t,i}(\mathbf{x})$ (red). Blue patch: intersection free region $\Gamma = \Delta_t \setminus \bigcup_{\mathbf{x}_e \in \operatorname{cone}_{t,i}(\mathbf{x})} B(\mathbf{s}_e)$. Dashed blue: Orthosphere \mathbf{s}_{\perp} of Δ_t and intersection free segments. $\hat{\mathbf{s}}_{\perp}$: advanced orthosphere of $\operatorname{cone}_{t,i}(\mathbf{x})$. Left: $\mathbf{x}_{\perp} \notin \bar{\Delta}_{t,i}(\mathbf{x})$, right: $\mathbf{x}_{\perp} \in \bar{\Delta}_{t,i}(\mathbf{x})$



Figure 4.5: Advanced orthosphere \hat{s}_{\perp} . Dashed segment: orthogonal projection $[p_j, p_k]$ of s_{\perp} onto face $[x_j, x_k]$ and resulting diametric orthosphere (dashed green circle). Hypothetical point $p \in B(s_e)$ of some external sphere s_e with $p \notin \hat{s}_{\perp}$.

which intersects s_{\perp} at either p_j , p_k or both, can intersect s_{\perp} at most orthogonal (Figure 4.5, red tangent segment). It follows geometrically that such a sphere s'_e and point p cannot exist and $\Gamma = \Delta_t \setminus B(s_e)$. s_e is unique by construction and called *advanced orthosphere* denoted $\hat{s}_{\perp}(i, t)$. It follows that $\min_{x_e \in \text{cone}_{i\perp}(\mathbf{s}), t \in \mathscr{R}}(d(s_i, s_e)) = d(s_i, \hat{s}_{\perp}(i, t))$



Figure 4.6: Corridor connectivity (a) Corridor of simplicial sphere world. s_{i_a} : agent sphere, x_g : agent goal. Red: clearance over cone estimates (b) Ordered sets of obstacles (white) and representative estimates Ω (grey circles) in left/right turns L_j , R_{j+1} and landmarks (red/blue dots) (c) Face clearance consists of four cone collision estimates.

With Proposition 1, the minimal clearance for the crossing a simplex t via the faces $\{i, j\}$ and $\{i, k\}$ is lower bound as:

$$d_{t,i} = \min\{d(s_i, s_j), d_c(t, i), d(s_i, s_k)\}$$
(4.32)

which consists of the estimate $d_c(t, i)$ and the clearance of the vertex spheres. The minimal clearance of transition of a sequence of simplices along a corridor can be expressed in terms of the combined collision distance $d_{t,i}$ of Proposition 1 for the transition across a face:

Definition 14 (Face clearance) For the shared face of two simplices $t_a = \{i, j, k\}, t_b = \{j, k, l\}$, with $t_a, t_b \in \mathcal{R}$, the clearance is defined as:

$$c(i,j) = \min\{d_{t_a,i}, d_{t_b,i}, d_{t_a,j}, d_{t_b,j}\}$$
(4.33)

The face clearance is illustrated in Figure 4.6 (c).

Lemma 4 (Collision free corridor) A planar navigation corridor, composed of an ordered list of k adjacent simplices $\mathbf{p} = (\mathbf{t}_i)_k \in \mathcal{R}$ with a start vertex $i_a \in \mathbf{t}_1$, for which \mathbf{s}_{i_a} is collision free and a goal point $\mathbf{x}_g \in \Delta_k$, with $\mathbf{x}_g \in \mathcal{F}$, is collision free with respect to a minimal clearance $c_{min} \in \mathbb{R}_{\geq 0}$, if the face clearance of all internal faces is bigger or equal to c_{min} .

Where \mathscr{F} is called *free space* and represents the accessible workspace of the polygonal sphere world $\mathscr{F} = \mathscr{W}_p \setminus \bigcup_i B(\mathbf{s}_i)$.

Proof. A center line, starting at x_{i_a} , crossing the corridor and ending at x_g , partitions the bounding vertices and cone distance estimates into a left and right hand set L and R of obstacles $t_{i,l}, t_{i,r}$ and collision estimates $\Omega(t_i)$ (Figure 4.6 (a)). Clearly, the corridor is collision free if $d(R, L) \ge c_{min}$. To relax this condition, observe that the corridor can be re-grouped into sub-sequences ..., $\mathcal{L}_j, \mathcal{R}_{j+1}, \ldots$ of adjacent cone transitions along the same *landmark* apex with opposite obstacles sequences ..., L_j, R_{j+1}, \ldots (Figure 4.6 (b)). Given landmark s_j and opposite set L_j , it is true that $d(R' = s_j, L_j) = \min_{t \in \mathcal{L}_j} \{d_{t,j}\} \ge \min_{t \in I_j} C_{j,k}\}_{k \in L_i}$. To estimate the minimal distance between opposite sets L_j and R_{j+1} , note that $s_j \in R_{j+1}$ and $s_{j+1} \in L_j$ and therefore it is true that:

$$\begin{aligned} d(L_{j}, R_{j+1}) &\geq \min\{d(\mathbf{s}_{j+1}, R_{j+1}), d(\mathbf{s}_{j}, L_{j}), d(\mathbf{s}_{j+1}, \Omega(\mathbf{t}_{L-R}, j+1)), d(\mathbf{s}_{j}, \Omega(\mathbf{t}_{R-L}, j))\} \\ &= \min\{d(\mathbf{s}_{j+1}, R_{j+1}), d(\mathbf{s}_{j}, L_{j}), d_{\mathbf{t}_{L-R}, (j+1)}, d_{\mathbf{t}_{R-L}, j}\} \\ &\geq \min\{d(\mathbf{s}_{j+1}, R_{j+1} \setminus \{\mathbf{s}_{j}, \Omega(\mathbf{t}_{L-R})\}), d_{\mathbf{t}_{L-R}, j}, d(\mathbf{s}_{j}, L_{j} \setminus \{\mathbf{s}_{j+1}, \Omega(\mathbf{t}_{L-R})\}), d_{\mathbf{t}_{R-L}, (j+1)}, d_{\mathbf{t}_{L-R}, (j+1)}, d_{\mathbf{t}_{R-L}, j}\} \\ &= \min\{d(\mathbf{s}_{j+1}, R_{j+1} \setminus \{\mathbf{s}_{j}, \Omega(\mathbf{t}_{L-R})\}), d(\mathbf{s}_{j}, L_{j} \setminus \{\mathbf{s}_{j+1}, \Omega(\mathbf{t}_{L-R})\}), c(i, (j+1))\} \end{aligned}$$

where $t_{L-R} \in \mathscr{L}$ is the simplex that is adjacent to $t_{R-L} \in \mathscr{R}$ and $\Omega(t_{L-R}, j+1)$) is the largest possible set $\Omega = \bigcup_{x_e \in \text{cone}_{t_{L-R}, j+1}(\mathbf{x})} s_e$. Note that $d(s_j, L_j \setminus \{s_{j+1}, \Omega(t_{L-R})\}) \ge d(s_j, L_j)$ and thus it follows that the minimal face clearance: $\min_{i=1,...,k-1} \{c(i, i+1)\}$ is a lower bound of the clearance of the corridor $(t_i)_k \in \mathscr{R}$.

95

4.4 High-level planning in simplicial sphere world complexes

4.4.1 Motion graph

The dual graph $G(\mathscr{R})$ of the regular triangulation (Definition 9) is called *power diagram*. Here, $G(\mathscr{R})$ is extended to serve as an abstraction of the sphere world for path planning.

Definition 15 (Face center) The center of the modified sphere \mathbf{x}'_c , i.e. the orthogonal projection of the orthosphere onto a face (Proposition 2), is called *face center* $\mathbf{x}_f(\mathbf{t}_i \cap \mathbf{t}_j)$ of the shared face of simplices $\mathbf{t}_i, \mathbf{t}_j$.

The face center is also the intersection point of the chordale of two vertex spheres with the edge.

Definition 16 (Face Center Graph) The line graph of $G(\mathscr{R})$ is called *face center graph* $FG = (V_f, E_f)^9$. The nodes V_f are located at the face centers.

An edge $e \in E_f$ connects two face centers and can be interpreted as the travel path across a simplex via the two faces.

Definition 17 (Motion Graph) Given an agent vertex $i_a \in \mathbf{s}$, and a navigation goal $x_g \in \mathcal{F}$, the *motion graph*

$$FG_m = (\{V_a, V_f\}, \{E_a, E_f\}, W)$$
(4.34)

is a weighted version of the face center graph FG with additional nodes $V_a = \{x_{i_a}, x_g\}$ and edges

$$E_r = \{ (\mathbf{x}_{i_a}, \mathbf{x}_f(\mathbf{t}_r \cap \mathbf{t}_j)) \} \cup \{ (\mathbf{x}_g, \mathbf{x}_f(\mathbf{t}_g \cap \mathbf{t}_j)) \},$$
(4.35)

with $t_a = \{t \in \mathcal{R}, i_a \in t\}$ and $t_g = t \in \mathcal{R}, x_g \in \Delta_{t_a}$ and $t_j = \{t \in \mathcal{R}, |t \cap t_{a,g}| = n\}$. The weights *W* are a function of the euclidean node distance $W_{ij} = f(||x_{V_i} - x_{V_j}||)$.

⁹A line graph $L(G) = (V_l, E_l)$ is an extension of some graph G = (V, E), where a new node is introduced for every edge in *E* and edges between nodes in V_l are introduced where two edges in E_f are incident to the same vertex in *V*.
An example of a motion graph is illustrated in Figure 4.8, left. To allow continued planning in an obstructed environment, edges with insufficient clearance are not removed from the motion graph. Instead, the *clearance penalty* function $f_c(x) : \mathbb{R} \to [0,\infty)$ is used to scale the euclidean edge weight in the case of insufficient clearance or even obstacle overlap. $f_c(x)$ is a monotonically increasing function of the robust edge overlap $x(e) = c_{min} - w(e)$, with $c_{min} \in \mathbb{R}_{\geq 0}$, some required minimal clearance, typically the agent diameter with a safety margin for object padding, if desired, and $w(e) = d(B(s_{e(0)}), B(s_{e(1)}))$, the sphere width along the edge *e*. Here, we use the clearance over the cone (Proposition 1) as an estimate for the width and an exponential with growth rate $r_c \in (0, 1]$ for f_c :

$$f_c(x) = \begin{cases} r_c^{-x} - 1, & \text{if } x \ge 0\\ 0, & \text{otherwise} \end{cases}$$
(4.36)

 r_c is called *clearance penalty* rate. For $r_c \rightarrow 0$, the penalty for insufficient clearance is big and concerned edges are effectively ignored in the motion graph. For $r_c \rightarrow 1$, the penalty vanishes and the volume of the obstacles is ignored. The concept is illustrated in Figure 4.7. The edge weight is then defined as:

$$W(e) = \left(1 + f_c(c_{min} - w(e))\right) \cdot \|\mathbf{x}_f(e(0)) - \mathbf{x}_f(e(1))\|$$
(4.37)

The clearance over the cone of Proposition 1 is a composite of observed clearances of the simplex faces and estimates across the opposite face. The conservatism in using it as a measure for the transition across a simplex can be relaxed, where needed, by further extending the clearance over the cone estimate into the adjacent simplex of the opposite face.

We use Dijkstra's algorithm to find a shortest path on the motion graph. The resulting path $p^* = (e_i) \in FG_m$ can be translated into the equivalent *navigation corridor*, the ordered list of adjacent simplices $\mathbf{t}^* = (t_i)_k \in \mathcal{R}$.



Figure 4.7: Penalty factor $f_c(x)$ for clearance x, high (solid) and low (dashed). Dashed circle: projection of robust minimal clearance c_{min} on a bottleneck of two obstacles.



Figure 4.8: Left: polygonal sphere world (grey) and motion graph (dashed) at time step t_0 , Black: Simplicial complex, Blue: agent sphere s_{i_a} with navigation path, Green: goal x_g . Right: Motion graph at t_1 . White area: navigation corridor (Δ_i), Cyan: identified path.

4.4.2 Path re-identification

With the next timestep and the obstacle positions $\mathbf{x}(t+1)$, the simplicial sphere world $\mathscr{R}(\mathscr{S}(t+1))$ is updated and a new optimal corridor $\mathbf{t}^*(t+1)$ is computed. The previous path $\mathbf{p}^*(t)$ is retained and used to re-identify a sub-region $\overline{\mathscr{R}} \subset \mathscr{R}(\mathbf{x}(t+1))$ as:

$$\overline{\mathscr{R}} = \{ \mathbf{t} \in \mathscr{R}(\mathbf{x}(t+1)) \mid \exists \ e \in \mathbf{p}^*(t), \Delta_t \cap e \neq \emptyset \}$$

$$(4.38)$$

In other words, $\overline{\mathscr{R}}$ are the simplices of $\mathscr{R}(\mathscr{S}(t+1))$ which are intersected by the previous path $p^*(t)$. The identified path $\overline{p^*}$ is the optimal path on the sub-region \overline{t} . Both paths, identified and optimal, have a cost *C*, associated with the sphere world at time step *t*. The cost difference can be interpreted as cost improvement if one path is chosen over the other. The relative cost difference is defined as:

$$\delta C = \frac{C(\overline{\mathbf{p}^*}) - C(\mathbf{p}^*)}{C(\overline{\mathbf{p}^*})}$$
(4.39)

To prioritize differing path segments closer to the agent, the edge cost is recalculated with progressively attenuated clearance penalty for distant segments:

$$C(\mathbf{p}) = \sum_{i=1}^{k} (1-\alpha)^{i} \cdot f_{c}(e_{i}) + d(e_{i}), \quad \alpha \in [0,1)$$
(4.40)

Attenuation, as defined in Equation (4.40), is applied to path solutions a posteriori, only for the purpose of comparing costs according to Equation (4.39) and not during the graph search. This is to avoid extending paths excessively where the attenuation is strong. The associated cost of changing the planned path is defined by the re-scaled cosine similarity (S_c) between directions from the agent coordinates to the first face centers of the two paths:

$$\delta\theta = \left(1 + S_c(\overline{\mathbf{p}^*}[0] - \mathbf{x}_{i_a}, \mathbf{p}^*[0] - \mathbf{x}_{i_a})\right)/2 \tag{4.41}$$

For exactly opposite pointing paths: $\delta \theta = 0$, for aligned directions: $\delta \theta = 1$.

The path decision heuristic for temporally persistent planning is defined as a function of the decision threshold $\delta \in [0, 1)$:

$$p_{p} = \begin{cases} p^{*}, & \text{if } \delta C \cdot \delta \theta > \delta \\ \\ \hline p^{*}, & \text{otherwise} \end{cases}$$
(4.42)

The heuristic relates the improvement in terms of relative path cost difference with the associated change of direction between the identified and optimal path. For $\delta = 0$, the optimal path is selected consistently, while for $\delta \rightarrow 1$, the identified path is chosen with increased probability. Note that for the case where the first segments of both paths are identical, the decision in Equation (4.42) still effectively has to be taken in order to persist on a path or not.

4.5 Simulations

To demonstrate the effect of temporally consistent navigation we simulate a navigation scenario with multiple spherical unicycle agents $s_a = (x, r, \theta)$ and one with a single agent to navigate around spherical moving obstacles.

4.5.1 Local motion planner

We use a unicycle motion model and the *move-to-projected-goal* law of (Arslan and Koditschek, 2016) as a local motion planner to safely implement the path of the motion graph. The kinematics describing the unicycle motion are given as:

$$\dot{\mathbf{x}} = \boldsymbol{v} \left[\cos(\theta), \sin(\theta) \right]^T, \quad \boldsymbol{v} \in \left[-\boldsymbol{v}_{max}, \boldsymbol{v}_{max} \right]$$
(4.43)

$$\dot{\theta} = \omega, \quad \omega \in [-\omega_{max}, \omega_{max}]$$
 (4.44)

The forward-moving unicycle is defined as a special case of Equations (4.43)-(4.44) with $v \ge 0$. In addition, we consider a fully actuated agent:

$$\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x}), \quad \|\mathbf{u}(\mathbf{x})\| \le v_{max}$$
 (4.45)



Figure 4.9: Local planner of (Arslan and Koditschek, 2016) for a unicycle model. Center: oriented agent sphere $s_a = (x, r, \theta)$, rectangle indicates heading θ . Dashed red: powercell $P(s_a) = \mathscr{LW}(s)$. Solid red: local free space $\mathscr{LF}(s_a)$. Solid blue: planned path. Red dot: intermediate goal x^{*}. Blue dot: \bar{x}^* . Green dot: \bar{x}^*_v . Cyan dot: \bar{x}^*_ω . Grey: cells of regular triangulation.

Since $s \in s$, i.e. the agent sphere is used as a vertex in the triangulation, here the power cell of the agent represents the *local work space* $\mathscr{LW}(s) = P(s)$ introduced in (Arslan and Koditschek, 2016). $\mathscr{LW}(s_a)$ is eroded by the agent radius *r* to form the *local free space* \mathscr{LF} , a collision free neighborhood of the agent:

$$\mathscr{LF}(\mathbf{x}) = \left\{ q \in \mathscr{LW}(\mathbf{x}) \mid B(\mathbf{q}, r) \subseteq \mathscr{LW}(\mathbf{x}) \right\}$$
(4.46)

By the sub-orthogonality assumption, it is guaranteed that $\mathscr{LW}(s) \neq \emptyset$. For the simulation we further require that obstacles do not overlap, such that $\mathscr{LF}(s) \neq \emptyset$. This is achieved if all obstacles respect the motion law for collision free navigation described below. With the following definitions the move-to-projected goal law is formulated to steer an agent to some location $x^* \in \mathscr{W}_p$:

$$\begin{split} \overline{\mathbf{x}}^* &= \begin{cases} \mathbf{x}^*, & \text{if } \mathbf{x}^* \in \mathscr{LF} \\ \Pi_{\mathscr{LF}}(\mathbf{x}), & \text{otherwise} \end{cases} \end{split} \tag{4.47} \\ \overline{\mathbf{x}}^*_{\mathcal{V}} &= \begin{cases} \mathbf{x}^*, & \text{if } \mathbf{x}^* \in \mathscr{LF} \cap H_N \\ \Pi_{\mathscr{LF} \cap H_N}(\mathbf{x}), & \text{otherwise} \end{cases} \\ \overline{\mathbf{x}}^*_{\omega} &= \begin{cases} \mathbf{x}^*, & \text{if } \mathbf{x}^* \in \mathscr{LF} \cap H_G \\ \Pi_{\mathscr{LF} \cap H_G}(\mathbf{x}), & \text{otherwise} \end{cases} \end{aligned} \tag{4.49}$$

where $\Pi_C(q)$ denotes the metric projection of q onto a convex set *C* and:

$$H_N(\mathbf{x},\theta) := \{\mathbf{q} \in \mathcal{W}_p \mid [\cos(\theta), \sin(\theta)](\mathbf{q} - \mathbf{x}) = 0\}$$
(4.50)

is the straight line motion range, due to the nonholonomic constraint of the unicycle,

$$H_G(\mathbf{x}, \mathbf{x}^*) := \{\omega \mathbf{x} + (1 - \omega) \mathbf{x}^* \in \mathcal{W}_p \mid \omega \in \mathbb{R}\}$$

$$(4.51)$$

is the line segment of \mathcal{W}_p containing x and x^{*}. These concepts are illustrated in Figure 4.9. The move-to-projected goal law for a unicycle is given as:

$$\nu = -k \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}^{\mathrm{T}} (\mathbf{x} - \bar{\mathbf{x}}_{\nu}^{*})$$

$$\omega = k \operatorname{atan} \left(\frac{\begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}^{\mathrm{T}} (\mathbf{x} - \frac{\bar{\mathbf{x}}_{\omega}^{*} + \bar{\mathbf{x}}^{*}}{2})}{\begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}^{\mathrm{T}} (\mathbf{x} - \frac{\bar{\mathbf{x}}_{\omega}^{*} + \bar{\mathbf{x}}^{*}}{2})} \right)$$

$$(4.52)$$

$$(4.53)$$

and for a fully actuated agent as:

$$u(x) = -k(x - \bar{x}^*)$$
 (4.54)

102

The velocities u, *v*, *w* are clipped according to the actuation limits in Equations (4.43)-(4.45). In the simulations performed here, we use a discrete-time version of these laws as described in (Arslan and Koditschek, 2019) with a control gain $k \rightarrow \infty$, while retaining the velocity limits mentioned above. In a static world it can be shown that with Equations (4.52)-(4.54), an agent model described in Equations (4.43)-(4.44) asymptotically steers almost all configurations towards any feasible goal x* (Arslan and Koditschek, 2016). We use this model for all agents and moving obstacles to realize collision free movement in the simulation. In (Arslan and Koditschek, 2016), x* is static, here agents pursue the first waypoint of the planned path x*(*t*) = p_{*p*}[0], which is a function of the current sphere world $\mathcal{S}(t)$ that includes the agent sphere as well.

Moving obstacles pursue intermediate goal $x^*(t) = x_g(t)$ describing a random walk. To avoid clustering of the obstacles resulting in the solidification of the environment over time, each obstacle performs a collision free random walk with a bias towards its home position x_h :

$$\mathbf{x}_g(t) = (1 - \beta)\mathbf{X} + \beta \mathbf{x}_h \tag{4.55}$$

where $X \sim U(\pm v_{max}) \in \mathbb{R}^2$ is a uniform random variable and $\beta \in [0, 1]$ is the home position bias. $x_h = x(t = 20)$ is established by starting with a regular grid of obstacles and letting them walk randomly, i.e. with $\beta = 0$, for 20 time steps, after which the home position is locked.

With very persistent planning $(\delta \to 1)$ or low clearance penalty $(r_c \to 1)$, the agent might tend to pursue an obstructed path and simply wait until the obstacles give way. This happens quickly, as the obstacles move to a random goal $x^*(t+1) \in \mathscr{LF}$ according to the move-toprojected-goal law (Equation (4.54)) and therefore the probability is high that the obstacle moves away from the collision boundary of \mathscr{LF} . To penalize this trivial behavior, the obstacle speed of a given obstacle s_{obs} is reduced proportionally according to:

$$v_{max,obs} = \begin{cases} (v_{max} - v_{min})\frac{d_{min}}{r_{peri}} + v_{min}, & \text{if } d_{min} < r_{peri} \\ v_{max}, & \text{otherwise} \end{cases}$$
(4.56)

where $d_{min} = d(s_a, s_{obs})$ is the distance to collision of the agent to a given obstacle sphere, r_{peri} is the obstacles safety perimeter, in this simulation defined as $r_{peri} = 1.2r_{obs}$, and $v_{min} = v_{max}/10$ is the lower speed limit.

4.5.2 Evaluation metrics

The simulation output represents a piecewise linear trajectory of the agent $\mathbf{x}_a = [\mathbf{x}(t=0), \dots, \mathbf{x}(t_{end})]$, describing the motion of the agent until reaching its goal: $\mathbf{x}(t_{end}) = \mathbf{x}_g$.

• The *consistency* of the agent movement is characterized with the orthogonal projection of a trajectory segment $\Delta x(t) = x(t) - x(t-1)$ onto the unit vector, indicating the direction of the previous segment $e(t-1) = \Delta x(t-1)/||\Delta x(t-1)||$:

$$\Delta \mathbf{x}(t) \cdot \mathbf{e}(t-1) \tag{4.57}$$

High consistency implies that the agent advanced well and in the same direction as for the previous time step. The consistency is low if either the advancement is small or orthogonal to the previous movement. For consistency, the average over an navigation run is reported.

• The *curvature* of a coordinate triple of the trajectory is calculated as the inverse radius of the circle through the coordinates:

$$\min\{\Delta\phi/\|x(t) - c(t-1)\|, 1\}$$
(4.58)

Where $\Delta \phi$ is the absolute angle between consecutive trajectory segments. It is upper bound by 1 to reject noise from trajectory segments with vanishing step size. For curvature, the average over an navigation run is reported.

• Interactions are counted as the set size of sphere labels that form the first face center of

the corridors $\mathbf{t}(t-1)$, $\mathbf{t}(t)$ of two consecutive time steps:

$$|(\mathbf{t}_1(t-1) \cap \mathbf{t}_2(t-1)) \cap (\mathbf{t}_1(t) \cap \mathbf{t}_2(t))|$$
(4.59)

- *Collisions* are defined in according to the slowing behavior of the obstacles described in Equation (4.56). A collision is registered in a timestep if the agent is within the obstacle perimeter.
- The total *corridor area* covered by all corridors t(*t*) during a navigation run from *t* = 0 until the goal is reached at *t_{end}* is defined as the union of all simplices ever involved in any corridor t(*t*):

$$\cup_{t=0}^{t_{end}} \{ \cup_{i \in \mathbf{t}(t)} \Delta_i(\mathbf{x}) \}$$

$$(4.60)$$

- The persistence *decision rate* is recorded as the rate of instances when the heuristic of Equation (4.42) is in favor of the identified path, when the first segment of the identified path is different from the optimal.
- The *total timesteps* until the goal is reached.
- The cumulative heading difference

$$\sum_{\tau=0}^{t} |\theta(\tau) - \theta(\tau - 1)| \tag{4.61}$$

is reported for unicycle agents.

4.5.3 Multiple agents with conflicting paths

For a qualitative demonstration of the effects of temporal persistence δ and clearance penalty r_c in relation with the actuation type, a scenario with eight agents in a circular workspace is used and repeated for fully actuated, and unicycle models. Initially, the agents are distributed evenly along a circle with goal coordinates locate opposite (Figure 4.10 (a)). A small, uniformly distributed perturbation $U(\pm \emptyset_{arena}/100)$ is applied to the coordinates to avoid perfect



symmetry. v_{max} is set to $\emptyset_{arena}/20/\Delta t$ and ω_{max} is limited to $\pi/8/\Delta t$ per time step.

Figure 4.10: Simulation results: fully actuated agents. (a) The total time steps until the goal is reached is improved with either a finite clearance penalty or persistent planning, but not if combined. Consistency is higher and less fluctuation if persistence is applied. Star denotes success in reaching goal. (b) Trajectories from start (circle) to goal, located opposite. With near infinite clearance penalty, erratic movements occur before the danger of collision is acute. Smoothness is improved in persistent planning. Combination leads more often to drastic turnarounds.

The simulation results are shown in Figures 4.10-4.12. For all motion models, the persistent planning allows to reduce the number time steps needed to reach the goal and the cumulative heading difference. The movement characteristics are improved, which manifests in higher and less fluctuating consistency. A finite clearance penalty allows to reduce erratic, oscillatory movement, especially in situations when collisions are anticipated, but not yet imminent. It can help to reduce the total time steps to reach the goal for fully actuated agents or the energy invested in heading alignment for unicycle agents. The effect is lost for forward-moving unicycle and leads to jamming if used in combination with persistence.

In conclusion, the non-holonomic constraint leads to a shift from erratic movements to increased investment in heading alignment, both of which are improved using persistent planning. The decline of performance when both strategies are employed arises when a plan, leading to a collision, is pursued persistently. This indicates the existence of a quantitative trade-off among the persistence parameter δ and the clearance penalty rate r_c , which is investigated in the scenario with a single agent in a cluttered dynamic environment.

4.5 Simulations



Figure 4.11: Simulation results: unicycle agents. Scenario according to Figure 4.10b. Star denotes success in reaching goal. (a) Analogous to, but less pronounced compared to fully actuated agent model scenario, the total timesteps until the goal is reached is improved with either a finite clearance penalty or persistent planning, but not if combined. Consistency is higher and less fluctuation if persistence is applied. (b) Cumulative heading difference is both reduced when either persistent planning or a finite clearance penalty. The improvement of the latter case is less apparent in terms of consistency, but more in terms of cumulative heading difference.



Figure 4.12: Simulation results: forward-moving unicycle agents. Scenario according to Figure 4.10b. Star denotes success in reaching goal. (a) Analogous to, but less pronounced compared to unicycle agent model scenario. Consistency is lower as agents decrease speed for turn-first movement in drastic directional change. (b) Cumulative heading difference generally higher than in the unicycle scenario. Combined use of persistent planning and finite clearance penalty leads to jamming.

4.5.4 Single-agent in a dynamic environment

To demonstrate the effect of persistent navigation of an agent in a complex dynamic world with unknown obstacle movement, we consider a hexagonal arena, populated with 122 or 144 randomly moving obstacle spheres. The agent is set to navigate to a static goal x_g across the arena. We choose five different experimental scenarios:

- 1. 122 obstacles, radius *r*_{obs}, area occupation: 79.4%
- 2. 144 obstacles, radius *r*_{obs}, area occupation: 82.4%
- 3. 122 obstacles, radius $1.2r_{obs}$, area occupation: 72.4%
- 4. Analogous to 1), with randomized heuristic

and perform a sweep of the δ decision threshold parameter over the entire range [0, 1) with n = 10 simulation runs for each configuration. A run is aborted after exceeding 1000 timesteps and reported regardless of the progress. Scenario 4) is used to evaluate the quality of the decision heuristic (Equation (4.42)). In scenario 4 the heuristic $h = \delta C \delta \theta$ it is randomized as:

$$h = P(X > \delta), \text{ with } X \sim U(0, 1),$$
 (4.62)

from a uniform distribution, i.e. the decision to persist with a plan is taken based on an random chance, regardless of the optimal and identified path cost.



Figure 4.13: Simulation results of four scenarios 1)-4) (row-wise) for navigation of single, fully actuated agent in cluttered, dynamic environment. Arena is equivalent to geometry shown in Figure 4.3, without polygonal obstacle. Box refers to first and third quartile, whiskers indicate range, orange line shows median, n = 10. Custom axis scales emphasized in red.

The results of scenarios 1)-4) correspond to the rows shown in Figure 4.13.

The baseline scenario 1 shows the the total timesteps until the goal is reached remain at a similar level for all values of δ , indication no overall loss of performance if persistence in increased. The consistency is optimal at ≈ 0.3 after which it declines and curvature increases. This effect is stronger in scenario 2), with more obstacles, and scenario 3) with the same amount but larger ones. This, and the more drastic decay of interactions in scenario 3), indicates that persistent planning has a bigger impact in an environment with lower clearances between obstacles than simply with a higher motion graph complexity.

The exponential decay of interactions comes at the price of moderate increase of collision situations for higher persistence. The total area covered decreases in a linear fashion, indicating that the path chosen remains geometrically restricted. The sensitivity of the decision rate is bigger at low values of δ and more so for complex environments.

Comparing scenario 1) to 4), shows that the positive effects of persistent planning in consistency, curvature, interactions, collisions and corridor area is lost in the range of $0 < \delta < 0.7$ and therefore validates the design of the heuristic in Equation 4.42. For $\delta \ge 0.7$, the drop in interactions, total corridor area and increase in collisions in scenario 4) represents an unreactive behavior, where the agent pursues a fixed path without regardless of the clearance. This behavior is equally present in all other scenarios.

4.5.5 Software

To construct the conforming constrained regular triangulation (Definition 8) we have built the python package ibrtpy¹⁰. A modified version of the incremental build algorithm introduced in (Edelsbrunner and Shah, 1996) for unconstrained triangulation is used that allows to conform to constraints. The algorithmic details are given in the Appendix A.4. To the best of our knowledge, it is the only open source library for constrained regular triangulations.

¹⁰https://c4science.ch/source/ibrtpy/

4.6 Discussion

This article presents a path planning algorithm for dense, dynamic and globally known environments. Together with a local planner that implements non-holonomic motion constraints, the algorithm generates temporally consistent and smooth paths and reduces collisions. The mesh used for the abstraction can represent polygonal boundaries and obstacles, and implicitly contains a property that is used to certify collision free corridors. If the spheres used to represent moving obstacles have equal radii, the triangulation mesh is equivalent to the a conforming constrained Delaunay triangulation¹¹. Delaunay triangulation can be built with standard software, while the certificates developed here remain valid. The heuristic presented allows to reassess previous path solutions, which can increase performance without making use of predictive techniques, and implicitly achieve a socially compliant robot behavior.

We identify the extension of the algorithm presented here to n = 3 dimensional sphere worlds as a promising solution for impersistent effects. Our results show that the algorithm displays more impressive performance in worlds with higher complexity, specifically with increasing number of dynamic obstacles as they result in a more complex triangulation. An additional dimension increases the number of edges and vertices of the motion graph simply because of the increased number of simplex neighbors. Proposition 1 is not directly extendable to higher dimensions as Lemma 2 does not generalize to higher dimensional simplices. However, as Proposition 2 holds in any dimension, we see the possibility of extending the concept of the advanced orthosphere for a higher dimensional estimate of the collision over the cone. Mesh generation is generally considered much harder in higher dimensions because certain simplex properties, like maximized minimal angle, which are desirable for applications such as finite element method simulations are more problematic to optimize. However, here we do not rely on any other property than the global simplex regularity, and thus it is expected to relax the difficulties of mesh generation in 3d.

¹¹To see this, consider the Definition of separating hyperplanes (Equation 4.10) that form the power cells (or *Voronoi* cells respectively). If $r_i = r_j$, $\forall s_i \neq s_j \in \mathbf{s}$, then H_{ij} is in invariant with respect to the actual values of r_i, r_j .

5 Discussion

Manipulative experiments are widely used in animal behavioral research to discover causal links between stimuli and responses. Compared to stationary systems, mobile agents can deliver stimuli directly to the selected individual for precise and minimally invasive manipulation. This capability is of particular interest for experiments with eusocial insects, whose life is governed by self-organization. In order to study natural social behavior the experiment must be performed within their nest environment.

In this thesis, the *Antbot* is presented as an instrument for manipulative experiments with ants. The hardware, consisting of nested manipulators that control a mobile dummy, is integrated with a state of the art tracking system that provides precise locations and postures of ants in real-time over weeks. We manufactured a robotic dummy with actuated synthetic antennae for tactile stimulation. We designed nests in way that the dummy can be exchanged without effort during or between experiments. A python software framework, FORTloop, is developed to facilitate the design of tracking experiments that require real-time data visualization and control of hardware. With FORTloop, complex concurrent programs can be implemented to realize the next generation Antbot that navigates the mobile dummy autonomously for automated manipulation.

Chapter 5. Discussion

The Antbot design process loosely follows the methodology introduced in (Mondada et al., 2013). In the first stage, tracking data of previous experiments were analyzed with respect to the very diverse characteristics of different ant species. Specifications for integration with the automated tracking system were considered. The goal is to achieve an acceptable level of universality to common research questions in ant behavioral research. To this end, we focused on the social factors that govern individual aggression, a fundamental behavior for colony cohesion and therefore for eusocial life. In the next stage, this question was investigated with the Antbot by running teleoperated manipulation experiments where ants were stimulated with touch in different social contexts. The successful completion of this biological experiment validates the basic framework.

In a fully crossed analysis, individuals need to be chosen according to a number of factors based on their identity. To study, for example, the influence of the activities individuals are occupied with prior to the stimulation as a factor on aggression, manipulations need to be balanced on the level of the identity and the activity status. This could be implemented with more sophisticated real-time data processing and visual aid to support the action planning of the operator, and therefore increase the degree of experimental automation. At the same time, autonomous navigation is a key component for which superhuman performance is expected. To implement autonomous navigation, we formulated a problem based on robot navigation in crowded dynamic environments with global knowledge on the world. A novel algorithm for socially aware navigation was developed, tested in simulations, and implemented in the Antbot framework. Finally, a demonstration was carried out to validate the algorithm for the realistic scenario at hand, under real-time computation constraints and using the full extent of the Antbot hardware.

The methodology introduced in (Mondada et al., 2013) follows a strict modeling approach, where the robotic system is used to validate a quantitative model, often formulated in analytic mathematical expressions. If this approach is successful, it represents the gold standard in understanding a self-organized phenomenon. In behavioral research on eusocial organisms, such models are difficult to obtain. This is even more true when the level of situatedness is high, i.e., the experimental design is parsimonious in controlling potential factors of importance. Here a - nevertheless hypothesis-driven - exploratory approach was taken, using a conservative experimental design with an exclusively tactile stimulus, that simplifies the interpretation of the response. Furthermore, studies can be observational and do not necessarily need to contain a mechanistic element.

5.1 Technical challenges

The autonomous operation of the Antbot presented in Chapter 2, Section 2.5 represents the start of the next design phase. The biggest challenge therein will be to program the Antbot to account for all possible situations that conflict with the experimental objective; e.g., ants climbing on top of the dummy, unreachable targets, and unforeseen technical issues that can occur over long term experiments. A frequently observed problem is the prevented retraction of the mobile dummy after an interaction because the security layer of the Antbot detects a collision with its interaction partner. In general, various situations may occur where the assumptions made during the development of the navigation algorithm presented in Chapter 4 will be violated. It is therefore necessary to develop new behaviors capable of resolving conflicts. Using the behavior tree introduced in Section 2.4.3, such solutions can be developed and tested very quickly. The hardware-in-the loop ('dryrun') mode of FORTloop (Section 2.4.2) provides an ideal test bench for the development the Antbot under realistic conditions.

The teleoperated experiment in Chapter 3 has shown that in order to enable fully crossed experimental design (Montgomery, 2017) at the individual level, depending on the factors to be considered, the manipulation can be extremely hard for the experimenter. For instance, to investigate whether the activity and social maturity of an individual (derived from the interaction network) can explain the probability for aggression, the same individual must be stimulated repeatedly while she is active and idle. Furthermore, individuals with different levels of the social maturity score must be selected and manipulated. Following such an experimental plan is challenging, and some form of deliberative action planning can address

this the challenge (Ghallab et al., 2016).

The concept of agents with learning capabilities in mixed societies has been outlined in (Landgraf et al., 2021). In such experiments, the robot learns its behavioral policy, i.e. the mapping of observations to robot actions on the fly according to some predefined scoring function. The authors claim that such a learning scheme may improve the social acceptance of the robot, and make the robot-animal interactions more realistic. While the benefit of such an approach for experiments with eusocial insects remains to be identified, the technical challenge is an interesting one.

5.2 Outlook

More complex research questions can be addressed with the existing teleoperated system. Colonies of *L. acervorum*, the model species used in this thesis, show synchronized short-term (15 min) activity cycles. Different theories have been proposed to explain the formation and utility of these activity cycles (Cole, 1991; Richardson et al., 2017). The Antbot platform is ideally suited to test these theories. As a second example, it has been found that the foraging behavior of the ant *Cataglyphis niger*, which does not depend on pheromone trails, is mainly governed by bumpy recruitment interactions where the speed of the counterpart influences the individual activity which in turn decides if an ant will leave the nest for food (Razin et al., 2013). This scenario seems realistic to be successfully investigated in a manipulative study with the tactile antbot dummy. Likewise, the capability of luring or simply displacing ants could be interesting with respect collective decision making for house hunting in *Temnothorax* ants, where recruitment and quorum sensing at a potential new site is a central mechanism (Sasaki et al., 2013). More general topics of interest include individual resting behavior (Helfrich-Förster, 2018) or camouflaging strategies with chemical signature and mimicked morphology, that are used for example by parasites (Akino, 2008; Fischer et al., 2020). The behavioral experiment described in Chapter 3 can thereby function as a baseline for more advanced experiments, targeting more complex perception, potentially on a higher level of mimicry, in terms of dynamics, body shape or chemical profile. The platform developed here provides the required versatility for extension in terms of hardware and software.

As a final remark, the Antbot represents an advanced platform for robotics research. It has been shown that ant species with very different movement characteristics are available for tracking experiments. As robots continue to be integrated in society, socially aware robot navigation has to be addressed swiftly. Experimental opportunities with human subjects however are sparse and therefore testing is often limited to simulations. The ant system could serve as a realistic test bench for low level testing of such algorithms in a complex, dynamic, noisy environments.

A Appendix

A.1 Antbot software architecture and implementation

A.1.1 Subprocess state machines

Figures A.1 and A.2



Figure A.1: Antbot hardware controller state machine.



Figure A.2: Antbot navigation process state machine.

A.1.2 Performance conisderations

Three major bottlenecks in the pyantbot software currently exist, accounting for almost 98% of the navigation process loop runtime:

- 1. Triangulation in ibrtpy.cibrt
- 2. Dual graph construction in tcnav.MotionGraph
- 3. Subregion identification in tcnav.MotionGraph

Bottleneck 1 is mostly caused by computationally expensive and ubiquitous set operations in python. Two solutions exist. First, the library could be rewritten in the C-language with moderate effort, allowing to benefit from better control over memory access. Second and probably intellectually more appealing, instead of building the triangulation from scratch in every iteration, it could be considered to update it only where necessary, i.e. where new vertices are added, moved or removed. The theory for such operations can be found in the literature (Cheng et al., 2013). Bottleneck 2 is caused by the fact that the dual graph and its linegraph are constructed top down, i.e. the entire graph of the triangulation is constructed, including all weights calculated based on the cone distance. This involves solving a 4 × 4 linear system of equation for every edge weight, in order to calculate the advanced orthocenter (Section A.4.4). While this is useful for plotting in debugging, it is absolutely unnecessary to construct the entire graph, as with more complex triangulations, the graph search will not propagate into a very large area. Some relaxation of this bottleneck is expected if the graph is constructed on-the-fly in a custom implementation of Dijkstra's algorithm. Bottleneck 3 is caused by a computationally intensive brute force implementation for identifying the subregion of the dual graph intersection by the navigation path of the previous iteration. Currently, the correct version of the code applies the Definition in Equation (4.38) for each segment and every simplex and then stores the simplices with non-empty intersection in a set. This causes the computational complexity to increase linearly with the number of simplices in the triangulation. In case of intensive triangulation refinement, as Defined in Section A.4.2,

Appendix A. Appendix

this causes the runtime to increase drastically and in addition, the standard deviation among iterations, as the number of simplices can vary drastically, depending on whether a sphere is located close to a border or not. Nevertheless, for the simulations in Chapter 4, the strict definition is applied. For the demonstration in Chapter 2, an alternative strategy is applied, which results minor inaccuracies. Instead of intersection waypoint segments, the two points of a segment are used to identify a triangle in which they are contained. Djikstra's is the used iteratively over all waypoint pairs to identify the shortest path on the dual graph. The set of simplices in this compound path is then used as subregion. This way, the runtime of the subprocess identification collapses to an average of < 1% of the brute force version and similar for the jitter, which is essential for a smooth agent movement. Note that the brute force method by no means the only strategy to identify the correct sub-region. A better way would be to explore adjacent simplices only for intersection, requiring the handling of some special cases, when waypoints are located exactly at the border or at vertices, for definite robustness.

A.2 Experimentation protocols

A.2.1 Tagging (without anesthetization)



(a) Materials



(b) Fixation film



(c) Step 1





(f) Step 4

(g) Step 5

(h) Step 6

Figure A.3: Tagging procedure. (a) From top left: tags, microscissors, insect pin and holder, felxible tweezers to handle ants, rigid tweezers to hold tag, toothpick, PATEX ULTRATMsuper glue, Fixation film from overhead projector (b). Tagging of *L. acervorum* ant: (c)-(e) Prepare fork and insert ant head first until thorax in slit. (f) Insert fork in slit to hold legs down. (g) Make sure thorax is well exposed in fork gap. (h) Apply glue and tag on thorax using toothpick. Let dry for > 5 min, then remove fork first and let ant escape by opening slit as shown in (c).



Figure A.4: Dummy fabrication. Left: Cutting the antenna with visual aid on petri dish (short laser engraved bar = 1mm, red arrow = 3 mm). Right: Dummy in fabrication fixation (counter magnet glued to underside of petri dish), with visual aid for angular antenna placement.

A.2.2 Robotic dummy fabrication

Figure A.4 illustrates the process.

- Materials: Small petri dish with engraved angle and distance ruler, un-covered magnet, gold coated magnet (see Table 2.1), synthetic brush hairs, super glue (Patex Ultra Gel), ceramic tweezers, plastic tweezers, binocular microscope, ethanol, incubator
- Protocol:
 - 1. Wash magnet, petri and tweezers dish with ethanol and let dry
 - 2. Cut hairs off brush (Lascaux LR 15060 6^{TM}) from the tip to account for different thickness and length. Length: 3 mm, and place in petri dish.
 - 3. Glue (Pattex ULTRA Gel[™]) hairs to robot according to "V" shape drawn on the underside of the dish in an angle of 60°. Let cure for 15 min.
 - 4. (If to be tracked) glue tag on robot on antenna side.
 - 5. Cure magnet in incubator with water reservoir for humidity for 4h at 80°.
 - 6. Wash ceramic tweezers with ethanol before manipulating robot.

A.2.3 Experiment preparation

Arena preparation

- Materials: un-fluoned box, nest spacer, clean IR pass glass, pate-a-fix, blue paper sheet, scotch, FluonTM, self adhesive Teflon film, credit card, sharp knife
- Protocol:

122

- 1. Take empty, un-fluoned box, make sure it has no scratches
- 2. Cover floor with Teflon film using credit card.
- 3. Cover bottom of box with blue paper.
- 4. Prepare large or small nest (2mm cardboard spacer with hidden door, no floor cardboard, IR pass filter cover. Make sure cover is clear. Place it using pate-a-fix.
- 5. Apply Fluon[™]to box walls.

Robot calibration

- Remark: this step differs for teleoperated snd autonomous experiment.
- Materials: Calibration arena, robot dummy (A.2.2)
- Protocol:
 - 1. Connect PC to robot hardware.
 - 2. Place robot in calibration arena.
 - 3. Launch keyboardantbot.
 - 4. Make sure that test arena is as close as possible to magnetic adapter and that adapter and arena are both at level (using the spirit. levels).
 - 5. Launch tracking system in test mode.
 - 6. Do test movements to establish θ_{offset} .
 - 7. (If closed loop operation intended) run calibration program to establish stagecamera coordinate transformation.
 - 8. Remove test arena.

Manipulator setup

- Remark: The system can be used in two modes:
 - *Teleoperation*: "open loop" guidance with the keyboard and visual feedback from the camera.
 - *Antbot programs*: "close loop", i.e. a target in pixel coordinates from the tacking system is used to guide the dummy automatically.
- Remark for usage with Antbot programs: The referencing depends on the camera focus. It is optimal to verify proper tag detection with a mock arena including tags and proper robot movement including oscillations (i.e. optimal height) before proceeding with this protocol. A referencing dummy with the appropriate height should be chosen to make sure it is detected without drastic camera focus adjustment. Every change on the camera focus afterwards will make the transformation slightly imprecise. The aperture position on has no effect.
- Manipulator stage start-up

- Prerequisites: Manipulator stage and controller set up, movement free to to full extent of the workspace.
- Protocol:
 - 1. Visually inspect manipulator, make sure it can move free in its entire range (potentially remove acrylic ant arena holder).
 - 2. Power tracking system, turn on PI controller
 - 3. Open the PI TERMINAL program and connect to controller
 - 4. Turn on servos: SVO 1 1 2 1 3 1
 - 5. Turn referencing for rotational axis off: RON 3 0
 - 6. Perform fast referencing of translation stages: FRF 1 2
 - 7. Place a tagged reference dummy on the magnet. Launch tracking to get a proper visual feedback from above. The reference dummy is a dummy with a spacer to place the tag at the same focal distance as the dummy that will later be placed on the ant arena.
 - 8. Move to well visible position: MOV 1 50.0 2 50. (the range is [0,50] for linear axes and infinite for the rotational one).
 - 9. By adjusting the position of the white 3d printed adapter with the two Torx screws, make sure that the dummy rotates around its center. As a hint: zoom in on the tracking stream on the dummy. Then place one corner of some window (e.g. PI Terminal) on the PC on a point on the dummy. Now rotate 360 degrees with MVR 3 360.0. Follow the trajectory of the "corner" of the open window on the image, it will describe a circle. Now place the corner of the window to the approximate center of the circle. Repeat the process until the corner of the middle is exactly in the center of rotation. The center should be coincident with the dummy center. If it is not the case, adjust the (white 3d printed) adapter position and repeat the process above until it is centered.
 - 10. Move the rotational axis to a horizontal position, pointing exactly to the right side (i.e. the front of the dummy) using MVR 3 dR (dR: some small number of degrees).
 - 11. Set the current rotational position to zero POS 3 0.0 (only possible if rotational stage was not referenced before), then check referenced state with FRF?. This should return "1" for all stages. Now the rotational stage is aligned and can perform relative and absolute moves.
 - 12. Move to corners of desired maximal travel range (to avoid crashing into the nest holder etc.) by vision and read out the coordinates with the MOV and POS? commands.
- References:
 - 1. PI Stage Controller manual: PIGCS_2_0_DLL_SM151E270.pdf
 - 2. Tracking system: https://github.com/formicidae-tracker/documentation/wiki
- Stage-camera coordinates calibration
 - * Protocol:
 - 1. Make sure the camera is focused and the stage is horizontal, as indicated by the spirit levels.

2. Run calibration script

python calibrate_stage.py hostname agent_tag on the correct host with the correct tag label of the dummy and change the maximal travel range in the beginning of the script (lolim, uplim, don't change the third number for the rotation range). The stage will perform a random walk with 20 waypoints to estimate the camera pixel to stage coordinates transformation.

- 3. The prediction error should be < 2mm for the stage (error sources: tag eccentricity, second order image distortion). The error for the angle should be well below 1°, since in the calibration configuration no friction is present. The prediction error for The output is a file named t_tostage_yyyy-mm-dd hh:mm:ss.sssss.npz that can be loaded by an experimental script.</p>
- Requirements:
 - PIPython must be installed in the current python environment: python \$HOME/../PIPython-1.3.5.37 setup.py
 - py-fort-hermes package must be installed:
 pip install formicidae-tracker-hermes
- Teleoperation
 - Protocol:
 - Open a terminal and start the keyboard antbot program: ./kAntbot hostname.config
 - Configuration file examples are located in kAntbot/configs. The following two options must be specified: name=highgarden, log_dir=... There are some more self-explanatory settings that change how the dummy moves.
 - 3. The program possesses some command line arguments that can be displayed with ./kAntbot -h. When using:

./kAntbot hostname.config --robot_frame the keys (see below) are used to move the dummy along its body axis. If this flag is not set, the keys are used to move the dummy along the stages axes. Movement in the robot coordinate frame is the standard usage. The other can be helpful for recovery.

- 4. A window with an ant inside should show up and the terminal should display system ready... This windows must always be the active window "the top window" in order for the keystrokes to be directed to the antbot program (and not for example to the internet browser).
- 5. To shut down, close this window.
- Key mapping:
 - * W/S Move forward/backward
 - * $\overline{A}/\overline{D}$ Move left/right
 - * (†)/(↓) Turn clock-/counterclockwise
 - * $(-)/ \rightarrow$ Quick 180° turn clock-/counterclockwise

- * 1, 2, 3, 4, 5 chose one of the frequency/amplitude presets (see config file)
- * Enter start oscillation
- * To be completed...
- Requirements:
 - 1. Keyboard antbot installed, repository with instructions can be found at: https://c4science.ch/diffusion/7833/
- Antobot programs
 - Prerequisites: same as for teleoperation, pyantbot Conda environment loaded, stage calibration done (see above), Intermediate processing done (see below)
 - Protocol:
 - * Launch a pyantbot program (see example follow_interact.py)
 - * For further information, see examples and code comments in *pyantbot* code base: https://c4science.ch/source/pyantbot/
 - Terminate: press q + Enter in terminal (terminates like any other FORTloop program)

Robot and ant colony installation

- Materials: Prepared arena including tagged colony (A.2.1), robot dummy (A.2.2
- Protocol:
 - 1. Start keyboardantbot. Do stage referencing if necessary.
 - 2. Place ant box.
 - 3. Verify that arena is still at level with adapter (A.2.3).
 - 4. Start tracking in test mode.
 - 5. Adjust optics and verify sufficient tag detection quality.
 - 6. Center ant box on camera.
 - 7. Move magnetic manipulator to verify that entire nest can be reached.
 - 8. Move magnetic manipulator approximately to hidden door.
 - 9. Place robot dummy. Perform small movements to verify functionality.
 - 10. Close box and start climate.

Intermediate processing

- Purpose: Generation of myrmidon metadata file to allow real time tag orientation and downstream processing such as interaction detection (actually only collisions in the current software state) in FORTloop (Figure 2.1).
- Prerequisites: System set up as described in (A.2.3) and running for a few hours to allow aggregation of sufficient tracking data for tag alignment.
- Protocol: Normal post-processing of preliminary tracking data in fort-studio.

Dummy insertion

- Materials: System set up as described in (A.2.3).
- Protocol:
 - 1. Wait two days after completing step (A.2.3).
 - 2. Check robot functionality with keyboardantbot.
 - 3. Open hidden door.
 - 4. Insert robot with keyboardantbot.
 - 5. Close hidden door.
 - 6. Wait 1h until starting experiments for colony to calm down in case of disturbance.

Experiment termination

- 1. Remove robot through hidden door and place in labeled petri dish
- 2. Stop experiment according to FORT standard.

A.3 Definitions for mesh construction

In the following Subsections A.3.1, A.3.2, A.3.3, definitions of essential concepts from Cheng et al. (2013) are reproduced and referenced. These definitions are essential for the modified construction algorithm for conforming regular triangulation given in Appendix A.4.

A.3.1 Simplex, face and facet

Definition 18 (Definition 1.3, Cheng et al. (2013)) A *k*-simplex \triangle is the convex hull of a set **x** of k + 1 affinely independent points. In particular, a 0-simplex is a *vertex*, a 1-simplex is an *edge*, a 2-simplex is a *triangle*, and a 3-simplex is a *tetrahedron*. A *k*-simplex is said to have *dimension k*. A *face* of σ is a simplex that is in the convex hull of a nonempty subset **x**. Faces of σ come in all dimensions from zero (vertices of σ) to k; \triangle is a face of \triangle . A proper face of \triangle is a simplex that is the convex hull of a proper subset **x**; i.e. any face except \triangle . In particular, the (k - 1)-faces of \triangle are called *facets* of \triangle ; \triangle has k + 1 facets. For instance, the facets of a tetrahedron are its four triangular faces.

A.3.2 Piecewise linear complex

Definition 19 (Definition 2.8, Cheng et al. (2013)) In the plane, a *piecewise linear complex* (PLC) \mathscr{P} is a finite set of linear cells - vertices, edges, and polygons - that satisfies the following properties:

- The vertices and edges in *P* form a simplicial complex. That is, *P* contains both vertices of every edge in *P*, and the relative interior of an edge in *P* intersects no vertex in *P* nor any other edge in *P*.
- For each polygon f in \mathcal{P} , the boundary of f is a union of edges in \mathcal{P} .
- If two polygons in \mathcal{P} intersect, their intersection union of edges and vertices in \mathcal{P} .

Note that in this paper we require that polygons f do not overlap. Edges in \mathcal{P} are called 128

segments, as opposed to an edge of a simplex which is called *face* or *facet* (see A.3.1). In the text, we denote a polygon with *P*.

A.3.3 Quarantined complex

Definition 20 (Definition 7.6 Cheng et al. (2013)) Let \mathscr{P} be a piece-wise linear complex in \mathbb{R}^n , let $S \subset |\mathscr{P}|$ be a finite point set that includes the vertices of \mathscr{P} , and let ω be a weight assignment such that a sub-complex Q of $DelS[\omega]$ is a *Steiner triangulation*¹ of \mathscr{P} . The dimension k of both \mathscr{P} and Q is the dimension of their highest-dimensional cell, which is not necessarily n. A j-simplex in Q is called a boundary simplex of Q if j < k and it is included in a linear j-cell in \mathscr{P} . We call Q a quarantined complex if it satisfies the following conditions:

- (i) The dimension of aff|Q| is equal to the dimension of Q.
- (ii) Every vertex in *Q* has nonnegative weight.
- (iii) The power distance between every pair of vertices in *Q* is nonnegative.
- (iv) For every boundary simplex σ of Q and every vertex v in Q, the power distance between $v[\omega_v]$ and the diametric ball $B(\sigma)$ of σ is nonnegative, i.e. $\pi(v[\omega_v], B(\sigma)) \ge 0$.

Remarks:

- $v[\omega_v]$ represents a weighted point, if interpreted as a sphere: centered at v with radius $\sqrt{\omega_v}$.
- |*Q*| is the *underlying space* of complex *Q*, i.e. the point-wise union of all cells in *Q* (Definition 1.8 in Cheng et al. (2013)). *af f*|*Q*| is the affine hull of |*Q*|, the set of all affine combinations of points in |*Q*|.
- A Steiner triangulation of some piece-wise linear complex \mathscr{P} (Definition 2.12 in Cheng et al. (2013)) is also called *conforming triangulation* of \mathscr{P} , a triangulation where the

¹Accroding to Definition 6

constraints of \mathscr{P} are respected and is potentially augmented by additional vertices on the constraints (called *Steiner points*).

- Note that for conditions (iii) and (iv), the power distance between two weighted spheres needs to be defined slightly different than Definition 3, i.e: $\pi(s_i, s_j) = ||x_i x_j||^2 r_i^2 r_j^2$. In the context of Definition 2, for two sub-orthogonal spheres it follows: $\pi(s_i, s_j) \ge 0$.
- Note that the conditions on the sphere world, as introduced in Section 4.2.3, inherently satisfy conditions (i), (ii) and (iii) is the required sub-orthogonality, which results in no redundant spheres. Condition (iv) is required for correctness of Proposition 1.

A.4 Mesh construction algorithm

A.4.1 Construction of constrained regular triangulations

In this paragraph the customization of an algorithm for building the constrained regular triangulation of a polygonal sphere world is described. Essential Definitions are given in Appendix A.3. As a convention, in this paragraph, in dimension \mathbb{R}^n , *n*-simplices are denoted by σ and n-1-simplices by f. The expression $f \in \sigma$ means that f is a face of σ , if not stated differently. For convenience, the expression f[i] represents some unspecified n-2-simplex in the (unordered) set of facets of f. A homogeneous simplicial *n*-complex \mathcal{T} , also called *triangulation*, is a set of simplices, in which every simplex face is also in \mathcal{T} and every non-empty intersection of two simplices is a face of both simplices. The word *homogeneous* requires that every simplex in \mathcal{T} , of dimension lower than *n*, is again a face of some *n*-simplex in \mathcal{T} . Intuitively, a triangulation divides the space into non-overlapping triangles and whereby the union of the triangles cover the entire space.

In computational geometry, the regular triangulation is usually defined more general for a point set $\{x_i\}$ with associated weights $w_i \in \mathbb{R}$ instead of spheres, without further specifications on how the weights are obtained. With the conversion $w_i = r_i^2$, weights or squared radii can be used interchangeably and yield the geometric interpretation of orthogonality (Definition 2). In

sphere worlds, we interpret the weights as squared sphere radii and, i.e. $r \in \mathbb{R}_{\geq 0} \leftrightarrow w \in \mathbb{R}_{\geq 0}$. To generalize the regular triangulation, it is useful to define the point lifting operation $lift(x, r^2)$: $\mathbb{R}^n \to \mathbb{R}^{n+1}$ that maps every weighted point $(x, r^2), x \in \mathbb{R}^n$ by adding a new coordinate² to its lifted representation $x^+ \in \mathbb{R}^{n+1}$:

$$x_{n+1} = \sum_{i=1}^{n} x_i^2 - r_i^2 \in \mathbb{R}^n$$
(A.1)

$$\mathbf{x}^{+} = [\mathbf{x}; x_{n+1}] \in \mathbb{R}^{n+1}$$
(A.2)

It can then be shown that the problem of constructing a regular triangulation is equivalent to finding the underside of the *lifted complex*, the convex hull of the lifted point set $\mathbf{x}^+ = {\mathbf{x}^+}_i$, and projecting it back onto the original space. This operation is illustrated in Figure A.5, right. Any triangulation that can be constructed in this way is called a *regular triangulation* Cheng et al. (2013). For vanishing radii the Delanuay case is obtained where all points \mathbf{x}^+ lie on the surface of a paraboloid centered at the origin. By adding a positive radius or weight to a point, it follows from Equation (A.1) that its lifted coordinate is lowered. Intuitively, increasing the weight, increases region of influence of a point on the triangulation. This way, the regular triangulation can encode short range vertex interactions. In Section 4.3 this property is used to derive a clearance estimate for navigation.

If a point x^+ is surrounded by points with sufficiently large weight, it can lie in the interior of $conv(x^+)$. Such points are called redundant³ and do not appear in the simplicial complex⁴. In the strict sense of Definition A.3.2, the resulting simplicial complex obtained by the lifting and projection operation is not a triangulation of the input vertex set **x** if redunant points exist. If all points have positive weight and are sub-orthogonal, it can be shown that no point is redundant, which motivates the assumption on the sphere world in Section 4.2.3. For unconstrained regular triangulation, the lifting analogy works without further restrictions and the use of efficient convex hull algorithms is beneficial.

²Also called *Lifted Companion* Cheng et al. (2013) or *Lifted Coordinate* Edelsbrunner and Shah (1996) ³Also called *submerged* or *hidden*.

⁴In the Delaunay case, $w_i = 0$, the lifted coordinates lie on a paraboloid and therefore no redundant points can exist.

Appendix A. Appendix

An important observation following from the lifting analogy is the fact that there exist exactly two well defined ways to triangulate any set of n + 2 points, if the lifted complex has a unique projection⁵: one is regular, i.e. the lower side of the lifted polytope, the other, which corresponds to the upper side, is not Edelsbrunner and Shah (1996). The transformation operation between the two is called a *flip* and is illustrated for n = 2 in Figure A.5, left. Two simplices sharing a link facet collectively represent a simplicial complex $\mathcal{T}(\mathbf{x})$ of the point set with $|\mathbf{x}| = n + 2$. \mathcal{T} can be called *locally regular* if it is regular. It follows that a triangulation is globally regular, if all complexes defined by the link facets are locally regular. Instead of building the triangulation in a top down approach using a convex hull algorithm, the *flipping*or *incremental build algorithm* is a bottom-up approach in which points are sequentially inserted in $\mathcal{T}(\mathbf{x}_{i-1})$ while the regularity of $\mathcal{T}(\mathbf{x}_{i-1} \cup (x_i, w_i))$ is maintained recursively through binary decisions on local regularity and flipping operations where appropriate Edelsbrunner and Shah (1996).

Two types of flips exist: insertion-deletion flips or flips that change shared face among simplices (Figure A.5, left). With the lifting analogy, establishing local regularity of a simplicial complex, consisting of n + 2 weighted points, amounts to check if a point (y, r_y^2) , that is not part of the simplex $\sigma(\{x, w\}_i)$, is below the surface spanned by the lifted points x_i^+ of σ . This can be done with the following predicate Cheng et al. (2013):

$$d = \det \begin{bmatrix} 1 & \mathbf{x}_{1}^{T} & \|\mathbf{x}_{1}\|^{2} - w_{1} \\ \vdots & \vdots \\ 1 & \mathbf{x}_{n+1}^{T} & \|\mathbf{x}_{n+1}\|^{2} - w_{n+1} \\ 1 & \mathbf{y}^{T} & \|\mathbf{y}\|^{2} - w_{y} \end{bmatrix}$$
(A.3)

If d > 0, then σ is locally regular with respect to the weighted point (y, w_y). The strict equality (d = 0) is avoided if the point coordinates are linear independent, which can be enforced in a

⁵This is the case if no more than *n* points have a lifted representations in \mathbb{R}^{n+1} which lie on a vertical hyperplane. To satisfy this condition, it is sufficient if these points have linear independent coordinates in \mathbb{R}^n .
consistent way through symbolic perturbations of the sphere coordinates Edelsbrunner and Mücke (1990). The flip operation itself is described in Edelsbrunner and Shah (1996) and is omitted here. Note that Equation (A.3) is sufficient to establish regularity and to this end, it is not mandatory to explicitly compute orthospheres as introduced in Definition 7.

Constraints that do not inherently respect regularity, appear as non-convex regions in the lifted complex. Therefore the convex hull algorithm cannot be used. With the flipping algorithm, constraints already present in the triangulation can enforced by preventing flips that cut through a constraint during point insertion Shewchuk (2003). Since the vertices that form the constraints have zero weight, we first construct a constrained Delaunay Triangulation of the polygonal workspace represented by the PLC \mathcal{P} , ignoring all spheres. For this purpose, a number of algorithms are available Cheng et al. (2013). Then the weighted points are introduced sequentially with the flipping algorithm. The sphere insertion procedure is described by Algorithm A.4.2).

Algorithm 1: *InsertPoint* algorithm of Edelsbrunner et al. Edelsbrunner and Shah (1996), adapted to maintain initial the constraints from a PLC *P*. *l.n.r.*: locally non regular

, c	<i>,</i>		
i	nput : $\mathcal{R}, \mathcal{P}, s$		
output:R			
10	$\sigma \leftarrow n$ -simplex in \mathscr{R} that contains s;		
2 if s not redundant w.r.t. σ then			
3	$links_{new} \leftarrow insertion(\mathscr{R}, s);$		
4	while $links_{new} \neq \emptyset$ do		
5	$link \leftarrow links_{new}$.pop;		
6	if <i>link l.n.r.</i> and <i>flippable</i> and $\notin \mathscr{P}$ then		
7	$links_{new}$.push(flip($\mathcal{R}, link$));		

As outlined above, the idea of Algorithm 1 is to use the lifting analogy on the smallest local region, i.e. n+2 points and perform flips as illustrated in Figure A.5. First, the point s is inserted into the current triangulation. At this point, s could be redundant, and its lifted version x^+ above the surface spanned by the lifted vertices of σ . In practice, this cannot happen as it is assumed that all spheres are sub-orthogonal. Therefore, in 2D, three new link facets will be produced and pushed on the *linksnew* stack. This stack is then processed in lines (4-7). Every

link facet involves two *n*-simplices σ_a, σ_b . If $\sigma_a \cup \sigma_b = conv(\sigma_a \cup \sigma_b)$, then the link facet is flippable with flip of type 2 \rightarrow 2. If this is not the case, the facet is flippable only the vertex in the interior has degree 3, which makes it a 3 \rightarrow 1 flip (see Figure A.5, center). In line 6 it is checked whether flippable link facets are that are not constrained by the polygonal workspace \mathscr{P} are regular or not and flipped if necessary. Algorithm 2, lines 1-3 inserts spheres sequentially into \mathscr{R} , starting from a constrained regular triangulation in line 1. The remainder of Algorithm 2 is dedicated to the refinement necessary to turn the constrained regular triangulation into a conforming constrained regular triangulation (Section A.4.2).

Algorithm 2: Incremental build algorithm for conforming regular triangulation of a PLC \mathscr{P} and set of spheres $\{s_i\}$. The effect of refinement (lines 4-9) is illustrated in Figure 4.3.

```
input : \mathcal{P}, {s<sub>i</sub>}
    output: R
1 \mathcal{R} \leftarrow ConstrainedDelaunay(\mathcal{P});
2 for s \in \{s_i\} do
    \mathscr{R} \leftarrow InsertPoint(\mathscr{R}, \mathscr{P}, s)
4 segments \leftarrow f \in \mathscr{P};
5 while segments \neq \emptyset do
         f \leftarrow segments.pop;
6
         for \sigma \in \mathcal{R}, f \in \sigma do
 7
                s_{o,f} \leftarrow orthosphere(f);
 8
                \mathbf{s}_v \leftarrow \sigma \setminus \mathbf{f};
 9
               if s_v not sub-orthogonal to s_o then
10
                     \mathscr{R} \leftarrow InsertPoint(\mathscr{R}, \mathscr{P}, (\mathbf{x}_{o,f}, 0));
11
                     segments.push([f[0], x_{o,f}], [x_{o,f}, f[1]]);
12
```

A.4.2 Triangulation refinement

After the first part of Algorithm 2, until line 3, skinny triangles can appear close to constraint segments (Figure 4.3, left). It can be shown that for a quarantined complex, all orthocenters lie within the underlying subspace of the triangulation (*Orthocenter containment Lemma 7.7* in Cheng et al. (2013)), which essentially prevents skinny triangles. To turn \mathscr{R} into a quarantined complex, the boundary segments need to be refined such that for every segment f and sphere s, the diametric orthosphere $s_{\perp}(f)$ is sub-orthogonal to s, i.e. $||x - x_{\perp}(f)||^2 \ge r^2 + r_{\perp}(f)^2$ (Section



Figure A.5: Flips and lifting analogy. Left, top: flips in 2d. Figure adopted from Edelsbrunner and Shah (1996). Left, bottom: Flipping history of an initially not flippable facet (dashed segment), where $\sigma_i \cup \sigma_j \neq conv(\sigma_i \cup \sigma_j)$, and a vertex with degree > 3. Right: lifted complex of n + 2 weighted points with projected lower and upper sides. Red circle and arrow illustrate effect of larger weight on lifted coordinate.

9.3 in Cheng et al. (2013)).

The diametric orthosphere $s_{\perp}(f)$ of an edge is illustrated in Figure 4.5 by the dashed green circle For a segment with zero weight vertices, this is simply the diametric sphere, centered in at the midpoint. To enforce the orthosphere containment condition for quarantined complexes (Figure 4.3, right), Algorithm 2 refines constraint segments f after the sphere insertion in lines 4-12. The refinement is defined for the general case of weighted vertices in REFINE-procedure in Section 9.3 of Cheng et al. (2013). For a constraint segment f which is a facet the simplex σ , an additional sphere with zero weight is inserted at the orthocenter $x_{\perp}(f)$ if the vertex of σ , which is not part of f, is not sub-orthogonal to the orthosphere of $s_{\perp}(f)$. Each newly formed sub-segment is then further refined if necessary. Note that in this way, Remark 1 (regularity in crossections) is re-establish in case of violation thourgh constraints.

A.4.3 Implementation details

The triangle⁶ package is used to build a constrained Delaunay triangulation in Algorithm 2, which is transformed to a *history directed acyclic graph*, the central structure of the incremental build algorithm Edelsbrunner and Mücke (1990). It represents the hierarchical structure of

⁶https://rufat.be/triangle/

past flipping operations and is used for efficient point localization and to maintain the simplex adjacency. Algorithms 1 and 2 use a symbolic input perturbation strategy called *simulation of simplicity* Edelsbrunner and Mücke (1990) to handle affinely not independent point sets. This results in a reproducible output and ensures that the algorithm terminates even in the presence of vertex configurations with ambiguous regularity⁷. Simulation of simplicity requires the use of integer or fixed point arithmetic to ensure correctness. Here we use a fixed precision of four decimals, which is seven orders of magnitude lower than arena size.

A.4.4 Computation of the orthosphere

The center \mathbf{x}_{\perp} and radius r_{\perp} of the orthosphere $\mathbf{s}_{\perp}(\mathbf{s})$ of a spheres set $|\mathbf{s}| = n + 1$ in general configuration can be calculated efficiently using the orthogonality conditions:

$$\|\mathbf{x}_{\perp} - \mathbf{x}_{i}\|^{2} = 0, \quad i = 1, \dots, n+1$$
 (A.4)

$$\leftrightarrow 2x_i^T x_{\perp} - (\|x_{\perp}\|^2 - r_{\perp}^2) = \|x_i\|^2 - r_i^2$$
(A.5)

which can be arrange in a linear system of equations:

$$\begin{bmatrix} 2\mathbf{x}_{1}^{\mathrm{T}} & -1\\ \vdots & \vdots\\ 2\mathbf{x}_{n+1}^{\mathrm{T}} & -1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{\perp}\\ \|\mathbf{x}_{\perp}\|^{2} - r_{\perp}^{2} \end{bmatrix} = \begin{bmatrix} \|\mathbf{x}_{1}\|^{2} - r_{1}^{2}\\ \vdots\\ \|\mathbf{x}_{n+1}\|^{2} - r_{n+1}^{2} \end{bmatrix}$$
(A.6)

This can be solved for x_{\perp} , r_{\perp} , e.g. using Cramer's Rule. Note that for building refined regular triangulations, the orthospheres do not need to be calculated explicitly, as the predicate of Equation (A.3) is used to test regularity. However, the orthosphere needs to be calculated to establish the distance to collision in Proposition 1.

⁷Referring to a situation in which both configurations of a close-to-degenerate complex of n + 2 vertices are determined not regular, which leads to an infinite flipping loop.

A.5 Incremental build regular triangulation python package

The code repository of incremental build regular triangulation python package ibrtpy which was written for this thesis can be found at https://c4science.ch/source/ibrtpy/.

Since the problem of building (unconstrained) regular triangulations can be mapped to the well explored problem of finding the convex hull of a set of lifted points (Section A.4), the incremental build algorithm is less popular and only used in special cases. Constrained conforming regular triangulations represent such a case and to the best of our knowledge, no freely available software exists for this purpose. ibrtpy is again a special case, using a constrained *Delaunay* (or unweighted) triangulation an input⁸ and inserting weighted points sequentially. Finally, the triangulation is refined if needed, to obtain a conforming constrained regular triangulation (Section 4.2). The package is a implementation for 2D of the incremental build algorithm for regular triangulations described in (Edelsbrunner and Shah, 1996).

This package is intended for research and educational purpose without a rigorous emphasis on computational efficiency. It uses the predicates described in (Edelsbrunner and Mücke, 1990). This allows to generate a deterministic output even in the case of an input which is not in general configuration, i.e. with collinear points. However, correctness is conceptually only guaranteed with a pure integer input. In this case the 'large number' strategy of (Edelsbrunner and Shah, 1996) to represent infinite vertices, which is implemented here in the strict sense through replacement of infinite symbolic elements in a matrix of dimension n by the absolute value of the largest finite number of that matrix to the power of n + 1, is exact. For floating point inputs, this strategy can lead to numerical imprecision due to large number, whereas for integers, very large numbers are handled by the integer 'bignum' arithmetic of python. It is the user's responsibility to ensures that the input is pure integer. In python, floats propagate. Very repetitive python set operations are re-implemented for sets of size up to three to achieve a considerable speedup.

⁸In the ibrtpy package, constrained Delaunay triangulations are computed using the Triangle package: https://www.cs.cmu.edu/~quake/triangle.html

Bibliography

- Agrawal, S., Safarik, S., and Dickinson, M. (2014). The relative roles of vision and chemosensation in mate recognition of drosophila melanogaster. *Journal of Experimental Biology*, 217(15):2796–2805.
- Akino, T. (2008). Chemical strategies to deal with ants: a review of mimicry, camouflage, propaganda, and phytomimesis by ants (hymenoptera: Formicidae) and other arthropods. *Myrmecological News*, 11(8):173–181.

Annagiri, S. (2021). Diacamma, pages 325–330. Springer International Publishing, Cham.

- Arslan, O. and Koditschek, D. E. (2016). Exact robot navigation using power diagrams. In 2016 IEEE International Conference on Robotics and Automation (ICRA), pages 1–8. IEEE.
- Arslan, O. and Koditschek, D. E. (2019). Sensor-based reactive navigation in unknown convex sphere worlds. *The International Journal of Robotics Research*, 38(2-3):196–223.
- Aurenhammer, F. (1987). Power diagrams: properties, algorithms and applications. *SIAM Journal on Computing*, 16(1):78–96.
- Belta, C., Isler, V., and Pappas, G. J. (2005). Discrete abstractions for robot motion planning and control in polygonal environments. *IEEE Transactions on Robotics*, 21(5):864–874.
- Berman, S., Lindsey, Q., Sakar, M. S., Kumar, V., and Pratt, S. C. (2011). Experimental study and modeling of group retrieval in ants as an approach to collective transport in swarm robotic systems. *Proceedings of the IEEE*, 99(9):1470–1481.

- Beshers, S. N. and Fewell, J. H. (2001). Models of division of labor in social insects. *Annual review of entomology*, 46:413.
- Bonnet, F., Cazenille, L., Gribovskiy, A., Halloy, J., and Mondada, F. (2017). Multi-robot control and tracking framework for bio-hybrid systems with closed-loop interaction. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4449–4456. IEEE.
- Bonnet, F., Gribovskiy, A., Halloy, J., and Mondada, F. (2018). Closed-loop interactions between a shoal of zebrafish and a group of robotic fish in a circular corridor. *Swarm Intelligence*, 12(3):227–244.
- Bourke, A. F. (1991). Queen behaviour, reproduction and egg cannibalism in multiple-queen colonies of the ant leptothorax acervorum. *Animal Behaviour*, 42(2):295–310.
- Buczkowski, G. and Silverman, J. (2005). Context-dependent nestmate discrimination and the effect of action thresholds on exogenous cue recognition in the argentine ant. *Animal Behaviour*, 69(3):741–749.
- Burridge, R. R., Rizzi, A. A., and Koditschek, D. E. (1999). Sequential composition of dynamically dexterous robot behaviors. *The International Journal of Robotics Research*, 18(6):535–555.
- Cai, K., Wang, C., Cheng, J., De Silva, C. W., and Meng, M. Q.-H. (2020). Mobile robot path planning in dynamic environments: A survey. *arXiv preprint arXiv:2006.14195*.
- Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraula, G., and Bonabeau, E. (2001a). Self-organization in biological systems. In *Self-Organization in Biological Systems*. Princeton university press.
- Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraula, G., and Bonabeau, E. (2001b). Self-organization in biological systems. In *Self-Organization in Biological Systems*. Princeton university press.
- Carew, T. J., Pinsker, H. M., and Kandel, E. R. (1972). Long-term habituation of a defensive withdrawal reflex in aplysia. *Science*, 175(4020):451–454.

140

- Chapman, B. B., Thain, H., Coughlin, J., and Hughes, W. O. (2011). Behavioural syndromes at multiple scales in myrmica ants. *Animal Behaviour*, 82(2):391–397.
- Charbonneau, D., Hillis, N., and Dornhaus, A. (2015). 'lazy'in nature: ant colony time budgets show high 'inactivity'in the field as well as in the lab. *Insectes sociaux*, 62(1):31–35.
- Charbonneau, D., Poff, C., Nguyen, H., Shin, M. C., Kierstead, K., and Dornhaus, A. (2017). Who are the "lazy" ants? the function of inactivity in social insects and a possible role of constraint: inactive ants are corpulent and may be young and/or selfish. *Integrative and comparative biology*, 57(3):649–667.
- Cheng, S.-W., Dey, T. K., Shewchuk, J., and Sahni, S. (2013). *Delaunay mesh generation*. CRC Press Boca Raton.
- Cole, B. J. (1991). Short-term activity cycles in ants: generation of periodicity by worker interaction. *The American Naturalist*, 137(2):244–259.
- Colledanchise, M. and Ögren, P. (2018). *Behavior trees in robotics and AI: An introduction*. CRC Press.
- Conner, D., Choset, H., and Rizzi, A. (2009). Flow-through policies for hybrid controller synthesis applied to fully actuated systems. *Robotics, IEEE Transactions on*, 25(1):136–146.
- De Rosa, A., Castro, I., and Marsland, S. (2022). The acoustic playback technique in avian fieldwork contexts: a systematic review and recommendations for best practice. *Ibis*, 164(2):371–387.
- Dell, A. I., Bender, J. A., Branson, K., Couzin, I. D., de Polavieja, G. G., Noldus, L. P., Pérez-Escudero, A., Perona, P., Straw, A. D., Wikelski, M., et al. (2014). Automated image-based tracking and its application in ecology. *Trends in ecology & evolution*, 29(7):417–428.
- Di Caro, G. and Dorigo, M. (1998). Antnet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 9:317–365.

- Dorigo, M., Birattari, M., and Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39.
- Edelsbrunner, H. and Mücke, E. P. (1990). Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Transactions on Graphics (tog)*, 9(1):66–104.
- Edelsbrunner, H. and Shah, N. R. (1996). Incremental topological flipping works for regular triangulations. *Algorithmica*, 15(3):223–241.
- Fainekos, G., Kress-Gazit, H., and Pappas, G. (2005a). Temporal logic motion planning for mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 2020– 2025.
- Fainekos, G. E., Kress-Gazit, H., and Pappas, G. J. (2005b). Hybrid controllers for path planning: A temporal logic approach. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 4885–4890. IEEE.
- Falanga, D., Kim, S., and Scaramuzza, D. (2019). How fast is too fast? the role of perception latency in high-speed sense and avoid. *IEEE Robotics and Automation Letters*, 4(2):1884– 1891.
- Feder, H. J. S. and Slotine, J.-J. (1997). Real-time path planning using harmonic potentials in dynamic environments. In *Proceedings of International Conference on Robotics and Automation*, volume 1, pages 874–881. IEEE.
- Feinerman, O., Pinkoviezky, I., Gelblum, A., Fonio, E., and Gov, N. S. (2018). The physics of cooperative transport in groups of ants. *Nature Physics*, 14(7):683–693.
- Feurtey, F. (2000). Simulating the collision avoidance behavior of pedestrians. *Master's thesis, University of Tokyo, Department of Electronic Engineering.*
- Fiorini, P. and Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. *The international journal of robotics research*, 17(7):760–772.

- Fischer, G., Friedman, N. R., Huang, J.-P., Narula, N., Knowles, L. L., Fisher, B. L., Mikheyev,
 A. S., and Economo, E. P. (2020). Socially parasitic ants evolve a mosaic of host-matching and parasitic morphological traits. *Current Biology*, 30(18):3639–3646.
- Foka, A. F. and Trahanias, P. E. (2010). Probabilistic autonomous robot navigation in dynamic environments with human motion prediction. *International Journal of Social Robotics*, 2(1):79–94.
- Forel, A. (1874). Les fourmis de la Suisse: Systématique, notices anatomiques et physiologiques, architecture, distribution géographique, nouvelles expériences et observations de moeurs, volume 26. Druck von Zürcher & Furrer in Commission bei H. Georg.
- Forel, A. (1923). Le Monde Social des Fourmis, volume 3. Librairie Kundig, Editeur.
- Forstmeier, W. and Schielzeth, H. (2011). Cryptic multiple hypotheses testing in linear models: overestimated effect sizes and the winner's curse. *Behavioral ecology and sociobiology*, 65(1):47–55.
- Franks, N., Blum, M., Smith, R.-K., and Allies, A. B. (1990). Behavior and chemical disguise of cuckoo antleptothorax kutteri in relation to its hostleptothorax acervorum. *Journal of Chemical Ecology*, 16(5):1431–1444.
- Fulgenzi, C., Tay, C., Spalanzani, A., and Laugier, C. (2008). Probabilistic navigation in dynamic environment using rapidly-exploring random trees and gaussian processes. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1056–1062. IEEE.
- Gal, A. and Kronauer, D. J. (2022). The emergence of a collective sensory response threshold in ant colonies. *Proceedings of the National Academy of Sciences*, 119(23):e2123076119.
- Gal, A., Saragosti, J., and Kronauer, D. J. (2020). antrax, a software package for high-throughput video tracking of color-tagged insects. *Elife*, 9.
- Ghallab, M., Nau, D., and Traverso, P. (2016). *Automated planning and acting*. Cambridge University Press.

- Gonon, D. J., Paez-Granados, D., and Billard, A. (2021). Reactive navigation in crowds for non-holonomic robots with convex bounding shape. *IEEE Robotics and Automation Letters*, 6(3):4728–4735.
- Grangier, J., Le Breton, J., Dejean, A., and Orivel, J. (2007). Coexistence between cyphomyrmex ants and dominant populations of wasmannia auropunctata. *Behavioural processes*, 74(1):93–96.
- Griparić, K., Haus, T., Miklić, D., Polić, M., and Bogdan, S. (2017). A robotic system for researching social integration in honeybees. *PloS one*, 12(8):e0181977.
- Guerrieri, F. J. and d'Ettorre, P. (2008). The mandible opening response: quantifying aggression elicited by chemical cues in ants. *Journal of Experimental Biology*, 211(7):1109–1113.
- Guerrieri, F. J., Nehring, V., Jørgensen, C. G., Nielsen, J., Galizia, C. G., and d'Ettorre, P. (2009). Ants recognize foes and not friends. *Proceedings of the Royal Society B: Biological Sciences*, 276(1666):2461–2468.
- Halloy, J., Sempo, G., Caprari, G., Rivault, C., Asadpour, M., Tâche, F., Saïd, I., Durier, V., Canonge, S., Amé, J. M., et al. (2007). Social integration of robots into groups of cockroaches to control self-organized choices. *Science*, 318(5853):1155–1158.
- Helfrich-Förster, C. (2018). Sleep in insects. Annual review of entomology, 63:69-86.
- Heyman, Y., Vilk, Y., and Feinerman, O. (2019). Ants use multiple spatial memories and chemical pointers to navigate their nest. *Iscience*, 14:264–276.
- Hölldobler, B. and Wilson, E. O. (2010). *The leafcutter ants: civilization by instinct*. WW Norton & Company.
- Hölldobler, B., Wilson, E. O., et al. (1990). The ants. Harvard University Press.
- Holman, L., Head, M. L., Lanfear, R., and Jennions, M. D. (2015). Evidence of experimental bias in the life sciences: why we need blind data recording. *PLoS biology*, 13(7):e1002190.

144

- Huber, L., Slotine, J.-J., and Billard, A. (2022). Avoiding dense and dynamic obstacles in enclosed spaces: Application to moving in crowds. *IEEE Transactions on Robotics*.
- Jandt, J. M., Bengston, S., Pinter-Wollman, N., Pruitt, J. N., Raine, N. E., Dornhaus, A., and Sih, A. (2014). Behavioural syndromes and social insects: personality at multiple levels. *Biological Reviews*, 89(1):48–67.
- Kane, G. A., Lopes, G., Saunders, J. L., Mathis, A., and Mathis, M. W. (2020). Real-time, lowlatency closed-loop feedback using markerless posture tracking. *Elife*, 9:e61909.
- Kay, T., Liberti, J., McKenzie, S. K., Richardson, T. O., Weitekamp, C. A., La Mendola, C., Rüegg,
 M., Kešnerová, L., Szombathy, N., McGregor, S., Rominguier, J., Engel, P., and Keller, L. (2022).
 Social network position defines behavior and physiology among worker ants. *in press*.
- Keller, L. and Chapuisat, M. (2001). Eusociality and cooperation. *Encyclopedia of Life Sciences*, pages 1–9.
- Kisilevsky, B. S. and Muir, D. W. (1984). Neonatal habituation and dishabituation to tactile stimulation during sleep. *Developmental Psychology*, 20(3):367.
- Krause, J., Winfield, A. F., and Deneubourg, J.-L. (2011). Interactive robots in experimental biology. *Trends in ecology & evolution*, 26(7):369–375.
- Kress-Gazit, H., Fainekos, G. E., and Pappas, G. J. (2009). Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics*, 25(6):1370–1381.
- Krieger, M. J., Billeter, J.-B., and Keller, L. (2000). Ant-like task allocation and recruitment in cooperative robots. *Nature*, 406(6799):992–995.
- Kruse, T., Pandey, A. K., Alami, R., and Kirsch, A. (2013). Human-aware robot navigation: A survey. *Robotics and Autonomous Systems*, 61(12):1726–1743.
- Kruusmaa, M., Gkliva, R., Tuhtan, J., Tuvikene, A., and Alfredsen, J. (2020). Salmon behavioural response to robots in an aquaculture sea cage. *Royal Society open science*, 7(3):191220.

- Kühbandner, S., Modlmeier, A. P., and Foitzik, S. (2014). Age and ovarian development are related to worker personality and task allocation in the ant leptothorax acervorum. *Current Zoology*, 60(3):392–400.
- Lahti, D. C. (2015). The limits of artificial stimuli in behavioral research: the umwelt gamble. *Ethology*, 121(6):529–537.
- Landgraf, T., Gebhardt, G. H., Bierbach, D., Romanczuk, P., Musiolek, L., Hafner, V. V., and Krause, J. (2021). Animal-in-the-loop: Using interactive robotic conspecifics to study social behavior in animal groups. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:487–507.
- Landgraf, T., Oertel, M., Rhiel, D., and Rojas, R. (2010). A biomimetic honeybee robot for the analysis of the honeybee dance communication system. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3097–3102. IEEE.
- Langen, T. A., Tripet, F., and Nonacs, P. (2000). The red and the black: habituation and the dear-enemy phenomenon in two desert pheidole ants. *Behavioral Ecology and Sociobiology*, 48(4):285–292.
- LaValle, S. M. (2006). Planning algorithms. Cambridge university press.
- Lenoir, A., d'Ettorre, P., Errard, C., and Hefetz, A. (2001). Chemical ecology and social parasitism in ants. *Annual review of entomology*, 46(1):573–599.
- Liberti, J., Kay, T., Quinn, A., Kesner, L., Frank, E. T., Cabirol, A., Richardson, T. O., Engel, P., and Keller, L. (2022). The gut microbiota affects the social network of honeybees. *bioRxiv*, pages 2021–12.
- Lichtenthäler, C., Lorenz, T., Karg, M., and Kirsch, A. (2012). Increasing perceived value between human and robots measuring legibility in human aware navigation. In *IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pages 89–94.
- Maccaro, J. J., Whyte, B. A., and Tsutsui, N. D. (2020). The ant who cried wolf? short-term 146

repeated exposure to alarm pheromone reduces behavioral response in argentine ants. *Insects*, 11(12):871.

- Mavrogiannis, C., Balasubramanian, K., Gandra, S. P. A., and Srinivasa, S. S. (2022). Winding through: Crowd navigation via topological invariance. *IEEE Robotics and Automation Letters*.
- Mavrogiannis, C., Baldini, F., Wang, A., Zhao, D., Trautman, P., Steinfeld, A., and Oh, J. (2021). Core challenges of social robot navigation: A survey. *arXiv preprint arXiv:2103.05668*.
- McGregor, S. G. (2022). *Proximate Mechanisms of Task Allocation in Ants*. PhD thesis, Faculté de biologie et de médecine de l'Université de Lausanne.
- Mersch, D. P., Crespi, A., and Keller, L. (2013). Tracking individuals shows spatial fidelity is a key regulator of ant social organization. *Science*, 340(6136):1090–1093.
- Mitri, S., Wischmann, S., Floreano, D., and Keller, L. (2013). Using robots to understand social behaviour. *Biological Reviews*, 88(1):31–39.
- Modlmeier, A. P. and Foitzik, S. (2011). Productivity increases with variation in aggression among group members in temnothorax ants. *Behavioral Ecology*, 22(5):1026–1032.
- Modlmeier, A. P., Liebmann, J. E., and Foitzik, S. (2012). Diverse societies are more productive: a lesson from ants. *Proceedings of the Royal Society B: Biological Sciences*, 279(1736):2142– 2150.
- Mondada, F., Martinoli, A., Correll, N., Gribovskiy, A., Halloy, J. I., Siegwart, R., and Deneubourg,J.-L. (2013). A general methodology for the control of mixed natural-artificial societies.Technical report, Pan Stanford Publishing Singapore.

Montgomery, D. C. (2017). Design and analysis of experiments. John wiley & sons.

Oster, G. F. and Wilson, E. O. (1978). *Caste and ecology in the social insects*. Princeton University Press.

- Parmentier, T., Dekoninck, W., and Wenseleers, T. (2015). Context-dependent specialization in colony defence in the red wood ant formica rufa. *Animal Behaviour*, 103:161–167.
- Parr, C. L., Dunn, R. R., Sanders, N. J., Weiser, M. D., Photakis, M., Bishop, T. R., Fitzpatrick, M. C., Arnan, X., Baccaro, F., Brandão, C. R., et al. (2017). Globalants: a new database on the geography of ant traits (hymenoptera: Formicidae). *Insect Conservation and Diversity*, 10(1):5–20.
- Phillips, M. and Likhachev, M. (2011). Sipp: Safe interval path planning for dynamic environments. In 2011 IEEE International Conference on Robotics and Automation, pages 5628–5635. IEEE.
- Rajendran, H., Haluts, A., Gov, N. S., and Feinerman, O. (2022). Ants resort to majority concession to reach democratic consensus in the presence of a persistent minority. *Current biology*, 32(3):645–653.
- Razin, N., Eckmann, J.-P., and Feinerman, O. (2013). Desert ants achieve reliable recruitment across noisy interactions. *Journal of the Royal Society Interface*, 10(82):20130079.
- Richardson, T. O., Kay, T., Braunschweig, R., Journeau, O. A., Rüegg, M., McGregor, S., De Los Rios, P., and Keller, L. (2021). Ant behavioral maturation is mediated by a stochastic transition between two fundamental states. *Current Biology*, 31(10):2253–2260.
- Richardson, T. O., Liechti, J. I., Stroeymeyt, N., Bonhoeffer, S., and Keller, L. (2017). Shortterm activity cycles impede information transmission in ant colonies. *PLoS computational biology*, 13(5):e1005527.
- Richardson, T. O., Stroeymeyt, N., Crespi, A., and Keller, L. (2022). Two simple movement mechanisms for spatial division of labour in social insects. *Nature Communications*, 13(1):6985.
- Rimon, E. and Koditschek, D. (1992). Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518.
- Rios-Martinez, J., Spalanzani, A., and Laugier, C. (2015). From proxemics theory to sociallyaware navigation: A survey. *International Journal of Social Robotics*, 7(2):137–153.

- Rosenthal, R. and Rosnow, R. L. (2009). *Artifacts in behavioral research: Robert Rosenthal and Ralph L. Rosnow's classic books*. Oxford University Press.
- Ruiz-Guajardo, J. C., Grossenbacher, D. L., Grosberg, R. K., Palmer, T. M., and Stanton, M. L. (2017). Impacts of worker density in colony-level aggression, expansion, and survival of the acacia-ant crematogaster mimosae. *Ecological Monographs*, 87(2):246–259.
- Sasaki, T., Granovskiy, B., Mann, R. P., Sumpter, D. J., and Pratt, S. C. (2013). Ant colonies outperform individuals when a sensory discrimination task is difficult but not when it is easy. *Proceedings of the National Academy of Sciences*, 110(34):13769–13773.
- Schielzeth, H. (2010). Simple means to improve the interpretability of regression coefficients. *Methods in Ecology and Evolution*, 1(2):103–113.
- Schultheiss, P., Nooten, S. S., Wang, R., Wong, M. K., Brassard, F., and Guénard, B. (2022). The abundance, biomass, and distribution of ants on earth. *Proceedings of the National Academy of Sciences*, 119(40):e2201550119.
- Serpell, E. and Chaves-Campos, J. (2022). Memory and habituation to harmful and nonharmful stimuli in a field population of the sensitive plant, mimosa pudica. *Journal of Tropical Ecology*, 38(2):89–98.
- Shewchuk, J. R. (2003). Updating and constructing constrained delaunay and constrained regular triangulations by flips. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 181–190.
- Shi, Q., Ishii, H., Kinoshita, S., Takanishi, A., Okabayashi, S., Iida, N., Kimura, H., and Shibata, S.
 (2013). Modulation of rat behaviour by using a rat-like robot. *Bioinspiration & Biomimetics*, 8(4):046002.
- Sprenger, P. P. and Menzel, F. (2020). Cuticular hydrocarbons in ants (hymenoptera: Formicidae) and other insects: how and why they differ among individuals, colonies, and species. *Myrmecological news*, 30.

- Stowers, J. R., Hofbauer, M., Bastien, R., Griessner, J., Higgins, P., Farooqui, S., Fischer, R. M., Nowikovsky, K., Haubensak, W., Couzin, I. D., et al. (2017). Virtual reality for freely moving animals. *Nature methods*, 14(10):995–1002.
- Stroeymeyt, N., Grasse, A. V., Crespi, A., Mersch, D. P., Cremer, S., and Keller, L. (2018). Social network plasticity decreases disease transmission in a eusocial insect. *Science*, 362(6417):941–945.
- Sturgis, S. J. and Gordon, D. M. (2012). Nestmate recognition in ants (hymenoptera: Formicidae): a review. *Myrmecological news*, 16:101–110.
- Tinbergen, N. (1951). The study of instinct. Pygmalion Press, an imprint of Plunkett Lake Press.
- Trautman, P., Ma, J., Murray, R. M., and Krause, A. (2015). Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation. *The International Journal of Robotics Research*, 34(3):335–356.
- Ulrich, Y., Saragosti, J., Tokita, C. K., Tarnita, C. E., and Kronauer, D. J. (2018). Fitness benefits and emergent division of labour at the onset of group living. *Nature*, 560(7720):635–638.
- Uslu, F. E. (2022). *Robotic manipulation to investigate the physical principles of biological self-organization*. PhD thesis, Institut Génie Mechanique, École Polytechnique Fédérale de Lausanne.
- Van den Berg, J., Lin, M., and Manocha, D. (2008). Reciprocal velocity obstacles for real-time multi-agent navigation. In 2008 IEEE international conference on robotics and automation, pages 1928–1935. Ieee.
- Van Toll, W. G., Cook IV, A. F., and Geraerts, R. (2012). A navigation mesh for dynamic environments. *Computer Animation and Virtual Worlds*, 23(6):535–546.
- Villacorta-Atienza, J. A. and Makarov, V. A. (2013). Neural network architecture for cognitive navigation in dynamic environments. *IEEE Transactions on Neural Networks and Learning Systems*, 24(12):2075–2087.

- Wang, B., Liu, Z., Li, Q., and Prorok, A. (2020). Mobile robot path planning in dynamic environments through globally guided reinforcement learning. *IEEE Robotics and Automation Letters*, 5(4):6932–6939.
- Whitehouse, M. E. and Jaffe, K. (1996). Ant wars: combat strategies, territory and nest defence in the leaf-cutting antatta laevigata. *Animal Behaviour*, 51(6):1207–1217.
- Wilkie, D., Van Den Berg, J., and Manocha, D. (2009). Generalized velocity obstacles. In 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 5573–5578. IEEE.
- Worm, M., Kirschbaum, F., and von der Emde, G. (2017). Social interactions between live and artificial weakly electric fish: Electrocommunication and locomotor behavior of mormyrus rume proboscirostris towards a mobile dummy fish. *PLoS One*, 12(9):e0184622.

CV - Matthias Rüegg

Rue Georges-Auguste-Matile 28, 2000 Neuchâtel matthias.ruegg@epfl.ch, matruegg@gmail.com Nationality: Swiss, Born: 10.03.1990

Education

2018-2023	PhD at EPFL, Doctoral program of Robotics, Control and Intelligent, Systems
	Advisors: Prof. Selman Sakar (EPFL) & Prof. Laurent Keller (UNIL)
2015-2017	MSc Mechanical Engineering (Specialization: Automation and Control), Minor
	in Computational Science and Engineering at EPFL
2017	Master thesis at Harvard University with Prof. Robert J. Wood: "Embedding a
	Control Architecture into a Flapping-Wing MAV"
2010-2015	BSc Mechanical Engineering at ETHZ, Focus: structural mechanics
2009	Matura, Kantonsschule Olten

Publications

Manuscript in submission: **Rüegg, M.**, Motes Rodrigo A., Sakar, M.S. and Keller L. Robotassisted manipulation to disentangle behavioral plasticity and task specialization as factors for aggression in ants.

Manuscript in submission: **Rüegg, M.**, Arslan Ö., Sakar, M. S. Temporally persistent motion planning in dense dynamic environments for socially aware navigation.

Manuscript under review: Kay, T., Liberti, J., McKenzie, S. K., Richardson, T. O., Weitekamp, C. A., La Mendola, C., **Rüegg, M.**, Kešnerová, L., Szombathy, N., McGregor, S., Rominguier, J., Engel, P., and Keller, L. (2022). Social network position defines behavior and physiology among worker ants.

Mailand, E., Özelçi, E., Kim, J., **Rüegg, M.**, Chaliotis, O., Märki, J., Bouklas, N. and Sakar, M.S., 2022. Tissue Engineering with Mechanically Induced Solid-Fluid Transitions. Advanced Materials, 34(2), 2106149

Özelçi, E., Mailand, E., **Rüegg, M.**, Sakar, M.S. and Oates, A.C., 2022. Deconstructing body axis morphogenesis in zebrafish embryos using robot-assisted tissue micromanipulation. Nature Communications, 13, 7934.

Richardson, T.O., Kay, T., Braunschweig, R., Journeau, O.A., **Rüegg, M.**, McGregor, S., De Los Rios, P. and Keller, L., 2021. Ant behavioral maturation is mediated by a stochastic transition between two fundamental states. Current Biology, 31(10), 2253-2260.

Bibliography

Feb-Aug 2015	R&D Intern at Sensirion AG, Stäfa, Mobile Systems group.
Jul 2009-Jul 2010	Landscaping, Assistant gardener Lindehof Garten- und Kommunalar-
	beiten, Andreas Widmer

Work Experience

Languages

Mother tongue: German French & English: Fluent, spoken and written

Various

Teaching: Control Systems (ME-321), Dynamical Systems (ME-221), Analysis (MATH-101), five student projects Regular military service, Swiss Army (2013-2014)