THE UNIVERSITY *of* EDINBURGH

HoverBot: A Manufacturable Swarm Robot that Has Multi-Functional Sensing Capabilities and Uses Collisions for Two-Dimensional Mapping

Markus P. Nemitz

The University of Edinburgh
School of Engineering
Institute for Integrated Micro and Nano Systems

The Centre for Doctoral Training
Intelligent Sensing and Measurement

Supervised by:
Dr. Adam A. Stokes

A Thesis Submitted for the Degree of
Doctor of Philosophy

2018

Lay Summary of Thesis

THE UNIVERSITY of EDINBURGH

The lay summary is a brief summary intended to facilitate knowledge transfer and enhance accessibility, therefore the language used should be non-technical and suitable for a general audience. (See the Degree Regulations and Programmes of Study, General Postgraduate Degree Programme Regulations. These regulations are available via: www.drps.ed.ac.uk.)

Name of student:	Markus Nemitz	UUN	S1356601
University email:	m.nemitz@ed.ac.uk		
Degree sought:	Doctor of Philosophy	No. of words in the main text of thesis:	32000
Title of thesis:	HoverBot: A Manufacturable Swarm Robot that Has Multi-Functional Sensing Capabilities and Uses Collisions for Two-Dimensional Mapping		

Insert the lay summary text here - the space will expand as you type.

I developed manufacturable swarm robots that are able to detect collisions and map their environment by colliding with their surroundings. My work on swarm robotic technology pushes swarm robotics research towards studies on collision-dependent behaviours, a research niche that has been barely studied. Collision events occur more often in dense areas and/or large groups, circumstances that swarm robots experience. Large groups of robots with collision-dependent behaviours could become a research tool to help invent and test novel distributed algorithms, to understand the dependencies between local to global (emergent) behaviours, and more generally the science of complex systems. Such studies could become tremendously useful for the execution of large-scale swarm applications such as the search and rescue of survivors after a natural disaster.

Document control

K:\AAPS\D-AcademicAdministration\02-CodesOfPractice,Guidelines&Regulations\24-MainReferencesCopiesPolicies\01-Current\Assessment BOE SCC & Feedback\Forms\ThesisLaySummary	
If you require this document in an alternative format please email Academic.Services@ed.ac.uk or telephone 0131 651 4990.	Date last reviewed: 31.05.16

Everything should be
made as simple
as possible,
but not simpler.

Albert Einstein

Abstract: Swarm robotics is the study of developing and controlling large groups of robots. Collectives of robots possess advantages over single robots such as being robust to mission failures due to single-robot errors. Experimental research in swarm robotics is currently limited by swarm robotic technology. Current swarm robotic systems are either small groups of sophisticated robots or large groups of simple robots due to manufacturing overhead, functionality-cost dependencies, and their need to avoid collisions, amongst others. It is therefore useful to develop a swarm robotic system that is easy to manufacture, that utilises its sensors beyond standard usage, and that allows for physical interactions. In this work, I introduce a new type of low-friction locomotion and show its first implementation in the HoverBot system. The HoverBot system consists of an air-levitation and magnet table, and a HoverBot agent. HoverBots are levitating circuit boards which are equipped with an array of planar coils and a Hall-effect sensor. HoverBot uses its coils to pull itself towards magnetic anchors that are embedded into a levitation table. These robots consist of a Printed Circuit Board (PCB), surface mount components, and a battery. HoverBots are easily manufacturable, robots can be ordered populated; the assembly consists of plugging in a battery to a robot. I demonstrate how HoverBot's low-cost hardware can be used beyond its standard functionality. HoverBot's magnetic field readouts from its Hall-effect sensor can be associated with successful movement, robot rotation and collision measurands. I build a time series classifier based on these magnetic field readouts, I modify and apply signal processing techniques to enable the online classification of the time-variant magnetic field measurements on HoverBot's low-cost microcontroller. This method allows HoverBot to detect rotations, successful movements, and collisions by utilising readouts from its single Hall-effect sensor. I discuss how this classification method could be applied to other sensors and demonstrate how HoverBots can utilise their classifier to create an occupancy grid map. HoverBots use their multi-functional sensing capabilities to determine whether they moved successfully or collided with a static object to map their environment. HoverBots execute an "explore-and-return-to-nest" strategy to deal with their sensor and locomotion noise. Each robot is assigned to a nest (landmark); robots leave their nests, move n steps, return and share their observations. Over time, a group of four HoverBots collectively builds a probabilistic belief over its environment.

In summary, I build manufacturable swarm robots that detect collisions through a time series classifier and map their environment by colliding with their surroundings. My work on swarm robotic technology pushes swarm robotics research towards studies on collision-dependent behaviours, a research niche that has been barely studied. Collision events occur more often in dense areas and/or large groups, circumstances that swarm robots experience. Large groups of robots with collision-dependent behaviours could become a research tool to help invent and test novel distributed algorithms, to understand the dependencies between local to global (emergent) behaviours and more generally the science of complex systems. Such studies could become tremendously useful for the execution of large-scale swarm applications such as the search and rescue of survivors after a natural disaster.

Contents

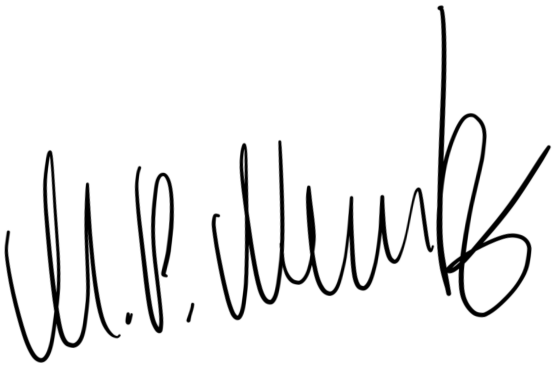
List of Figures

List of Tables

List of QR Codes

Declaration of Authorship

I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where states otherwise by reference or acknowledgment, the work presented is entirely my own. Any included publication is my own work, except where indicated throughout the thesis and summarised and clearly identified on the declarations page of the thesis.



I dedicate this dissertation to my father Peter Nemitz.

Acknowledgements

First, I would like to express my sincere gratitude to my advisor Dr. Adam A. Stokes for his continuous support of my studies and research, for his immense patience, motivation, and knowledge. I could have not imagined a better advisor and mentor for my doctoral studies. Besides my advisor, I would like to thank the director and co-director of the Centre for Doctoral Training in Intelligent Sensing and Measurement: Professor Andrew Harvey and Professor Ian Underwood, for their insightful comments and encouragement, but also for their hard questions which inspired me to widen my research from various perspectives. My sincere thanks also goes to Professor Edwin Olson and Professor Alfred O. Hero, who provided me with invaluable advice, who gave me the opportunity to join their research groups as academic affiliate at the University of Michigan, and who gave me access to their laboratories and research facilities. My research would have not progressed at this pace without their support. I thank my fellow colleagues for the stimulating discussions, sleepless nights of hard work, and for all the fun we have had in the past four years.

Last but not least, I would like to thank my family. Their constant inspiration and guidance kept me focused and motivated. I am highly grateful to my father Peter for giving brilliant advice in every aspect of my life. I cannot express my gratitude for my mother Bernadette in words, whose unconditional love has been my greatest strength. My profound thanks goes to my beloved brother Matthias, my sister-in-law Kathrin, my niece Lucy and my god-daughter Emma for their immense love and support throughout my studies. I would like to thank my wife Cher for her unconditional support. She has accompanied me throughout the entire process and made many sacrifices to maximise our time together when I studied in Edinburgh and she worked in Ann Arbor. She has made my life brighter and an exciting adventure ever since I met her on the Camino de Santiago in April 2013.

Chapter 1

Introduction

Like most robotics research, swarm robotics is a technology driven research field. Advances in robot technology usually enable new types of experiments which lead to new insights and discoveries. This thesis deals with the improvement of current state-of-the-art swarm robotics technologies for the general advancement of experimental research and for swarm research on collision-dependent behaviours.

Swarm robotics is the study of developing and controlling scalable groups of simple robots. Individual robots within a swarm only possess limited capabilities. They move in two- or three-dimensional space, sense their local environment, and communicate with their nearest neighbours. These local interactions between hundreds or thousands of robots can potentially give rise to complex behaviours. For example, a swarm of robots could perform a search-and-rescue mission. The entire swarm covers an area, robots locally interact with one another to find and retrieve survivors. An individual would be unable to perform such a complex behaviour. The goal of swarm robotics research is to substitute a few sophisticated robots with many simple robots to gain robustness, flexibility and to circumvent single-robot-failures from resulting in mission failure [1]. Applications range from space-exploration to finding survivors after large-scale natural disasters.

Figure 1.1: Examples of swarms in nature. From the top left to the bottom right: a swarm of honeybees, a swarm of termites, a flock of geese, a flock of starlings, a school of fish, and a herd of sheep [2]. I attach the permission to reuse this figure in Appendix D.3.

Almost all swarm robotics research is inspired by the observation of emergent behaviors in nature [3]. Colonies of termites collaborate to build termite mounds with integrated ventilation mechanisms which protect the colony from critical temperatures. Schools of fish cluster together to make it difficult for a visually orientated predator to pick and grab an individual before it disappears into the school thereby increasing chances of survival. Flocks of birds fly in formation and take turns in positioning to maximise the total travelled distance as a collective. Figure 1.1 shows a collection of swarms in nature. The control in such natural systems is entirely distributed among the individuals without having a leader that coordinates activities. These natural systems accomplish complex global tasks through simple local interactions of large groups of autonomous individuals and are commonly referred to as examples of swarm intelligence.

A lot of research in swarm robotics has been conducted via computer simulations. Brambilla et al. analyzed more than 60 publications that dealt with swarm robotic collective behaviors [1]. They found that more than half of these publications presented results which were obtained through simulations or models. Although simulators are a valuable tool for systematically exploring the algorithmic-behaviour of swarms, they frequently involve simplifications and reductionist axioms to enable computational tractability. Such simulated systems can fail to faithfully reproduce the intricate physical interactions and variability that exist in real systems, and their fidelity to the real world is difficult to verify or improve without feedback from physical experiments [4]: while computer simulations are constrained by imagination, physical experiments are constrained by hardware.

1.1 Motivation

Building hardware is a challenging task. Swarm robotics researchers frequently face a cost-functionality optimization problem when it comes to building scalable robot swarms. For example, every additional sensor on a robot increases the power consumption of the system, requires an additional sensor specific input on the microcontroller, requires additional space, and increases the overall cost. As a result, research in large-scale swarms ($\geq 1,000$) often sacrifices sensing, processing and locomotion capabilities for the size and quantity of robots, and these design decisions substantially limit the type of experiments that researchers can perform. Limiting robot functionality limits the depth of experiments and the depth of discoveries. Increasing robot functionality *usually* increases cost and therefore constrains swarm size.

Therefore, instead of increasing the quantity of robots in a swarm by reducing the functionality of each robot, the motivation for this thesis comes from the opportunity to develop a new swarm robotics technology that is scalable and delivers increased functionality at low cost. I identify and address three major swarm robotic challenges in this thesis: i) the manufacturing effort of swarm robots by developing a novel swarm robot that is unprecedentedly easy to manufacture and assemble, ii) the cost-functionality trade-off in swarm robotics by developing signal processing techniques that augment the sensing capabilities of robots, and iii) the lack of physical interactions between swarm robots and their immediate environment by demonstrating how robots can utilise collisions with their surroundings to map their environment.

1.2 Problem Definition

There are three problems that I tackle in this thesis: I address the manufacturing and assembly problem in Chapter 3, the functionality-cost problem in Chapter 4, and the lack of physical interactions problem in Chapter 5.

1.2.1 Manufacturing and Assembling Robots Is Expensive

There is a considerable manufacturing-assembly overhead for existing swarm robotic systems. Every component of a robotic system that requires manual assembly invokes labour cost. Manual labour by skilled-engineers limits the practicality of fabricating and experimenting with robot collectives at scale.

1.2.2 Cost Determines the Functionality of Robots

The functionality of a robot is related to its cost, whereas robot cost usually impacts swarm size, hence robot functionality determines swarm size. This is a problem since robot functionality must increase and robot cost retain to advance the types of experiments that can be performed.

1.2.3 Swarm Robots are Designed to Avoid Physical Interactions

The majority of swarm robotic systems avoid physical interactions such as collisions although their biological counterparts embrace them. Swarm roboticists possibly discard a phenomenon that could help understand complex systems, and a rich information source that contains insights about the environment and state of a robot collective.

1.3 Aims and Objectives

In accordance with the problem definition, the aims of this thesis are:

- to invent and demonstrate a swarm robotic system that is easily manufacturable, in which robots do not require manual assembly and do not avoid physical collisions.
- to develop a method that increases robot capability without hardware modifications.
- to start exploring collision dependent swarm behaviours.

The objectives of this PhD thesis are:

- To conduct a literature review on previous swarm robotic systems, to analyse their locomotion and sensing capabilities, and to elaborate what factors influence their manufacture most. Does the locomotion strategy of a robot have a greater impact on robot assembly than sensors?
- To develop a robot that only consists of a Printed Circuit Board (PCB) and surface mount components to eliminate the need for manual robot assembly. The battery is excluded from this expectation.
- To develop a framework that helps evaluate whether sensors have been used to their maximum potential. Are there sensors that make robots more functional than others? Can sensors be used beyond their standard functionality?
- To investigate why designers have made swarm systems avoid collisions and whether that is useful in swarm robotics research since the field is heavily inspired by colliding species. Why do biological swarm agents such as ants or bees collide and swarm robots do not? What do biological agents gain from collisions? Are there any prior swarm robotic studies that deal with robot collisions or utilises collisions for a task?

1.4 Preview of Contributions

- The systematic review of the locomotion strategies of previous swarm robotic systems and an analysis of their influence on robot manufacture [5].
- The development of a new low-friction locomotion strategy for swarm robots [5][6].
- The development of the HoverBot system [5][6].
- The systematic review of the sensing capabilities of previous swarm robotic systems and an analysis of the circumstances under which specialised sensors become multi-functional [7].
- The analysis of HoverBot's magnetic field measurements during movement and association to robot rotations, successful movements, and collisions [7].
- The modification of signal processing techniques to build and operate an online classifier on HoverBot's low-cost microcontroller to gain multi-functional sensing capabilities (detecting rotations, successful movements, and collisions) [7].
- The review of colliding systems in nature. The analysis of swarm robotic systems and whether they collide with their environment. The review of previous work on collision-dependent behaviours in swarm robotics [8].
- The development of an "explore-and-return-to-nest-after-n-steps" mapping strategy that utilises collisions with static objects to map an arena environment. The demonstration of four robots mapping their environment by varying the number of steps they take before returning to their nest (landmark) [8].

1.5 Publications

This thesis represents my own work, and includes a number of original contributions to scientific knowledge. The work presented in this thesis has led to two journal publications, one workshop publication, and a patent application filed by the University Court of the University of Edinburgh.

[J1] M. P. Nemitz, M. E. Sayed, J. Mamish, G. Ferrer, L. Teng, R. M. McKenzie, A. O. Hero, Edwin Olson, and A. A. Stokes, **HoverBots: Precise Locomotion Using Robots That Are Designed For Manufacturability**, *Frontiers in Robotics and Artificial Intelligence*. (2017) [5]

[J2] M. P. Nemitz, Ryan Marcotte, M. E. Sayed, G. Ferrer, A. O. Hero, Edwin Olson, and A. A. Stokes, **Multi-Functional Sensing for Swarm Robots Using Time Sequence Classification: HoverBot, an Example**, *Frontiers in Robotics and Artificial Intelligence*. (2018) [7]

[C1] M. P. Nemitz, and A. A. Stokes, **HoverBots: Embracing and Detecting Collisions Using Robots Designed for Manufacturability**, *International Conference on Robotics and Automation. Swarms: From Biology to Robotics and Back Workshop*. (2018) [8]

[P1] M. P. Nemitz and A. A. Stokes. **A Locomotion Platform and Multiagent System**. *Patent application*: GB1611448.0. (2016) [6]

During the course of my doctoral studies, in addition to my own work, I have also contributed to other colleagues' research. The following work has not been featured in this dissertation.

[J3] D. Ross, M. P. Nemitz, and A. A. Stokes. **Simulating Soft Robotic Systems: Insights from a Thermodynamic Perspective**. *Soft Robotics*, The Path Ahead, pp. 0-47. (2016) [9]

[J4] M. P. Nemitz, P. Mihayalov, W. T. Barraclough, D. Ross, and A. A. Stokes. **Using Voice Coils to Actuate Modular Soft Robots: Wormbot, an Example.** *Soft Robotics*, The Path Ahead, pp. 0-36. (2016) [10]

[J5] A. C. McConnell, M. Vallejo, R. C. Moioli, F. L. Brasil, N. Secciani, M. P. Nemitz, C. P. Riquart, D. W. Corne, P. A. Vargas, and A. A. Stokes. **SOPHIA: Soft**

Orthotic Physiotherapy Hand Interactive Aid. *Frontiers in Mechanical Engineering*. (2017) [11]

[J6] L. Teng, K. Jeronimo, T. Wei, M. P. Nemitz, G. Lyu, and A. A. Stokes. **Integrating Soft Sensor Systems Using Conductive Thread**. *Journal of Micromechanics and Microengineering*. (2018) [12]

[J7] S. T. Mahon, J. Roberts, M. E. Sayed, D. H. Chun, S. Aracri, R. M. McKenzie, M. P. Nemitz, and A. A. Stokes. **Capability by Stacking: The Current Design Heuristic for Soft Robots**. *Biomimetics*. (2018) [13]

[J8] L. Teng, K. Pan, M. P. Nemitz, R. Song, Z. Hu, and A. A. Stokes. **Soft RFID sensors: Wireless Long-range Strain Sensors Using Radio-frequency Identification**. *Soft Robotics*. (2018) *accepted*

[J9] R. M. McKenzie, M. E. Sayed, M. P. Nemitz, B. W. Flynn, and A. A. Stokes. **Linbots: Soft, Modular Robots Utilising Voice Coils**. *Soft Robotics*. (2018) *accepted*

[J10] M. E. Sayed, M. P. Nemitz, S. Aracri, A. C. McConnell, R. M. McKenzie, and A. A. Stokes. **The Limpet: A ROS-Enabled Multi-Sensing Platform for the ORCA Hub**. *MDPI Sensors*. (2018 - under review)

During the course of my doctoral studies, I was awarded awards, grants, and positions for my research and extra-curricular outreach.

Awards

Best Poster Award on
Collective Perception in Multi Agent Systems
Centre for Doctoral Training (CDT) in Intelligent Sensing and Measurement
Annual Conference 2015/16, United Kingdom

Global Impact at Home Shortlisted Nominee for
Establishing a Research Collaboration with the University of Michigan
Edinburgh University Students' Association (EUSA) 2016, United Kingdom

Best Oral Presentation Award on
Distributed Sensing with Swarm Robots
CDT in Intelligent Sensing and Measurement
Annual Conference 2016/17, United Kingdom

Grants

International Internship Grant (£4000) on
Distributed Sensing with Swarm Robots
at University of Michigan advised by Prof. Edwin Olson 2017, United States
CDT in Intelligent Sensing and Measurement

Positions

Certificate of Achievement for Acting as
President of the Engineering Graduate Society
from September 2015 to December 2016 at
the University of Edinburgh, School of Engineering, United Kingdom

Awarded the Position of
Academic Affiliate
from December 2015 to September 2018 at
University of Michigan, Department of Computer Science and Engineering
advised by Prof. Edwin Olson and Prof. Alfred O. Hero, United States

1.6 2D Barcodes

Throughout my thesis, I use 2D barcodes to direct the reader towards important videos that help understand the context and other thesis relevant videos such as the manufacturing video of HoverBots. I use Latex's *qrcode* package. It encodes data such as html links into two-dimensional barcodes. There is a list of all barcodes at the beginning of the thesis on page xviii. If the linked content asks for a password, please enter *phdthesis*. Most mobile phones such as the iPhone series are able to read the barcodes. Therefore, please open the standard camera application on you phone, aim for the barcode, and open the link that pops up. Otherwise I refer to *QR Reader for iPhone* in the AppStore or *QR Code Reader* in the Google Play Store; both applications are freeware.

Test your camera. Link to my webpage.

1.7 Thesis Outline

In Chapter 2, I give an overview of the field swarm intelligence. I introduce examples of biological swarm systems in Section 2.1.1, extract their system level properties in Section 2.1.2, and list the swarm level behaviours they are able to perform in Section 2.1.3. In Section 2.1.4, I bridge the gap to robotic systems and discuss a few examples. In Section 2.2, I review a broad list of swarm robotic systems for their locomotion capabilities. In Section 2.3, I review the same list of swarm robotic systems for their sensing capabilities. I talk about physical collisions in natural and robotic systems in Section 2.4.

I introduce the entire HoverBot system in Chapter 3. I will start by explaining low-friction locomotion in Section 3.1. In Subsection 3.1.1, I describe a robot in terms of a free body diagram to understand the force vectors that apply to robots. I introduce techniques that help reduce the friction between robots and their substrate in Subsection 3.1.2. In Subsection 3.1.3, I introduce low-friction locomotion with a magnet-levitation table. In the following Section 3.2, I describe the magnet-levitation table in detail. In Subsection 3.2.1, I give a brief overview over the magnet-levitation table iterations that I have built. I describe the fluid mechanics of the table in Subsection 3.2.2 and I explain the magnetic grid that I embedded into the top surface of the table in Subsection 3.2.3. In Subsection 3.2.4, I explain the final implementation of the magnet-levitation table. In Section 3.3, I introduce the HoverBot, a swarm robot that consists of a PCB and a battery. I explain my thoughts and considerations during the design process in Subsection 3.3.1. In Subsection 3.3.2, I give a brief overview over five different robot implementations that I have built during my studies. In the following Subsection 3.3.3, I explain my microcontroller choice and give a brief overview of the software that I programmed. In Subsection 3.3.4, I approximate magnetic fields of planar coils, discuss the design of planar coils, describe actuation circuits that are able to energise planar coils, and explain the coil actuation sequences that make HoverBots move. I discuss the sensors that I have implemented in Subsection 3.3.5, and the communication and visual outputs in Subsection 3.3.6. I explain the power system in Subsection 3.3.7 and the PCB design in Subsection 3.3.8. In Section 3.4, I explain the overhead camera system that I have used to observe HoverBots, and in Section 3.5, I describe the overhead infrared communication system

that I have used to read out sensor data and debug HoverBots. In Section 3.6, I demonstrate the system by letting robots perform a variety of behaviours.

In Chapter 4, I develop a signal processing technique that enhances HoverBot's sensing capabilities without changing its hardware: I start by studying the capabilities of HoverBot's magnetic field sensor in Section 4.1. From there, I describe three different magnetic field profiles that HoverBot measures when it i) successfully moves; ii) collides with a static object; or iii) accidentally rotates by 45 degrees. In Section 4.3, I describe the challenges in discriminating between these magnetic field profiles. In Sections 4.4 to 4.6, I explain how I use many of such magnetic field profiles as data to train a time series classifier. Each profile class has its own representation which I upload on HoverBot's microcontroller. HoverBot's classifier takes a new measurement after each movement and analyses it for each profile class and assigns it to the maximum-likelihood class. I indicate the classifier's performance in Section 4.7. I discuss my signal processing approach in Section 4.8, its applicability to other sensors and robots in Section 4.9, and its research applications in Section 4.10.

In Chapter 5, I start exploring collision dependent swarm behaviours by taking advantage of HoverBots' capabilities to embrace and detect collisions. I let a group of four HoverBots collide with their surroundings to collectively map their environment. I give an overview over the experimental setup in Section 5.1. In Section 5.2, I explain the mapping algorithm and in Section 5.3, the parameters that influence the mapping task. In Section 5.4, I show the outcome of my mapping algorithm: an occupancy grid map.

In Chapter 6, I reflect on my own work and make conclusions. I map out a wide range of future work in Chapter 7. In Appendix A, I discuss the technical details of my work. In Appendix A.1, I discuss the manufacture of the magnet-levitation table; in Appendix A.2, the manufacture of the tethered and untethered HoverBot prototypes; and in Appendix A.3 the manufacture of HoverBot.

Chapter 2

Background and Literature Survey

This chapter covers the background of, and literature related to, my thesis. In Section 2.1, I give a brief introduction to the field of swarm intelligence. I cover examples of biological swarms in Subsection 2.1.1, their high-level attributes in Subsection 2.1.2, the swarm behaviours they perform in Subsection 2.1.3, and a variety of physical embodiments (robots) in Subsection 2.1.4. I discuss the various locomotion strategies that have been implemented in previous swarm robotic systems in Section 2.2. I compare the locomotion strategies according to their cost and manufacturing effort. In Section 2.3, I evaluate the sensing capabilities of previous swarm robotic systems and elaborate whether some sensors increase the functionality of a robot more than others. In this section I also introduce the instrument model, a sensor model from the sensor community that helps understand under which circumstances sensors become multi-functional. In Section 2.4, I study swarm robotic systems from a physical interaction standpoint. I evaluate which swarm robotic systems are capable of colliding and detecting collisions and whether there have been swarm robotic studies on collision-dependent collective behaviours. I briefly look into nature to determine whether biological systems collide with their surrounding.

2.1 Swarm Intelligence

Swarm intelligence is a biologically driven research field. It originates from observations in nature. Large collectives of simple agents accomplish complex tasks through simple interactions. In the following sections, I will give several examples that have served as inspiration and motivation for the pursuit of research in swarm intelligence. If we uncover the underlying science of complex systems, applications will appear in every aspect of our lives ranging from the famine in third world countries to colonizing other planets. In Section 2.1.3, I introduce various swarm behaviours that depict active areas of research.

2.1.1 Biological Swarm Systems

2.1.1.1 Insect Colonies

Insects such as ants are considered eusocial insects, they show an advanced social organisation [14]. Ant collectives can consist of hundreds to thousands of individual ants, whereas each seeks out food, responds to environmental circumstances such as chemical signals of its peers, and fights intruders. These rather simple entities perform trivial tasks. However, as group they are able to build entire nests out of soil, leaves, and twigs that contain complex underground passages, and warm brooding chambers whose temperature is carefully controlled by decaying nest materials and the ants' own bodies [15]. Ants construct complex bridges with their own bodies to overcome structural gaps in their environment. They collaborate to build living bridges and provide a shortcut for foraging peers [16]. Although we know much about ants and their social structure, scientists still cannot explain their local-to-global behaviours and how evolution has produced creatures with such an enormous contrast between individual simplicity and collective sophistication[17].

Legionary ants attack a wasp nest

2.1.1.2 The Brain

Ant colonies and brains are very similar, they are both large-scale collectives of simple entities that give rise to complicated and sophisticated behaviours [18]. The individual entity in the brain is called a neuron. Neurons are specialised communication cells. They are excitable cells that bidirectionally transport information to and from the central nervous system. Neurons are linked to many other cells and it is generally believed that the complex network of neurons is responsible for perception, thought, feelings, consciousness, and other important large-scale brain activities [17].

2.1.1.3 The Immune Response

The adaptive immune response is comprised of individual decisions from individual cells based on information provided by the immediate neighbourhood and interpreted through membrane receptors. It is capable of cooperatively detecting and encountering sophisticated invaders. The adaptive immune response depicts nature's most successful large-scale sensing system. Lymphocytes (white blood cells) play a crucial role in the immune response. They can be viewed as agents migrating through the body searching for pathogens. If a lymphocyte finds a pathogen, it outputs large numbers of specialised antibodies that find similar invaders. The antibodies basically go on a search-and-destroy mission, whereas the lymphocyte is dividing into daughter lymphocytes to produce even more antibodies [19]. Just like the neurons of a brain and the ants of an ant colony, large groups of cells and their interactions are responsible for an emergent phenomenon, here the defence of the host against pathogens [17].

2.1.2 System Level Properties

The previous examples of biological collectives demonstrate distributed capability among large groups of simple agents. In the following, I extract the attributes that can be found in natural swarms [20], and explain why they are advantageous in robotics too.

2.1.2.1 Robustness

Large-scale collectives are robust to mission failures due to single-unit-errors. If a swarm loses parts of its collective, the chances are high that it can still continue the task at hand with the remaining agents. The loss of an individual can be immediately compensated. The simplicity of agents makes them also less prone to failure. Robustness is one of the main reasons for conducting swarm robotics research. Applications such as space exploration immensely profit from robustness; rather than sending a single expensive robot to Mars (Mars Rover), we could explore new planets with swarms of inexpensive robots, thereby increasing mission success in case of single robot errors.

2.1.2.2 Flexibility

Large collectives are able to coordinate their behaviours to execute a variety of tasks. In Section 2.1.3, I summarise various swarm behaviours. For example, ants can find the shortest path to a food source and build ant bridges to overcome gaps. Researchers envision assigning robots to tasks without telling them how to solve them. The advantage is that such systems are unpredictable since even the programmer does not know how the swarm is going to solve it.

2.1.2.3 Scalability

Swarms are able to operate in a wide range of group sizes. The control algorithms must be designed for very large swarm sizes; algorithms and hardware should be *scalable*. This property ensures that swarms can be increased in size "on the fly". Some applications such as search and rescue profit from scalability; the larger the swarm, the more area can be covered in a given amount of time potentially deciding between life or death.

2.1.3 Swarm Level Behaviours

There are a variety of swarm behaviours, applications usually require a combination of behaviours to be executed. For example, in a search and rescue scenario, *dispersion* in combination with *foraging* might be required to find survivors and *cooperative transport* to retrieve them. In the following, I briefly introduce the most important swarm behaviours.

2.1.3.1 Aggregation

Self-organised aggregation is the grouping of individuals into a cluster without input of external clues such as environmental modifications. These behaviours can act as precursors to other behaviours such as self-assembly. There have been several studies on aggregation [21] [22].

2.1.3.2 Dispersion

Self-organised dispersion is the opposite of self-organised aggregation. A group of individuals tries to increase coverage over an area; robots try to maximise the distance to each other while maintaining in communication range [23] [24] [25]. This behaviour is useful in applications such as surveillance.

2.1.3.3 Foraging

Foraging is the process of a swarm searching for the best food source, whereas best is defined as the source that maximises the ratio of returned food to spent resources such as number of individuals involved [26] [27] [28].

2.1.3.4 Self-Assembly

Self-assembly is the self-organised assembly of individuals to structures such as ant bridges or rafts. In these cases, self-assembly can help the collective overcome obstacles or survive floods, scenarios that they cannot overcome as individuals [16] [29] [30].

2.1.3.5 Connected Movement

Connected movement is the assembly of individuals to achieve collective movement. For example, ants use tandem running to lead other ants from the nest to food sites. During a tandem run, the follower ant keeps in contact with the leading ant touching its legs and abdomen [31]. This problem has been studied with robots; robots connect via physical connectors which helps them overcome environmental challenges such as gaps [32] [33].

2.1.3.6 Cooperative Transport

Large collective of individuals cooperate to transport objects such as food. This behaviour has been often observed in ants in which groups of ants collectively transport prey, prey that none of the individuals could carry by itself. They coordinate their pulling and pushing actions to create an overall larger moving force.

2.1.3.7 Pattern Formation

Pattern formation deals with the formation of geometric or functional shapes. In geometric pattern formation, a large collective assembles to a shape through local interactions amongst individuals similar to the formation of crystals. In functional pattern formation, the pattern of the collective is dictated by the environment. For example, the shape of ants that carry prey is determined by the local environment and the dimensions of the prey. Notable robotic work on pattern formation includes Rubenstein et al. work who experimented with a 1024-unit robot swarm [4] to generate star, H-, and other arbitrary patterns.

2.1.3.8 Self-Organised Construction

Self-organised construction can be observed in many social insects such as in termites. Termites build complex termite mounds without having a single unit in charge. Termites make local adaptations to their nest, whereas the termite mound is an emergent phenomenon. There have been several studies

on self-organised construction. Professor Kirstin Petersen from Cornell University dedicated her PhD thesis on this specific topic [34]. Her thesis depicts an excellent starting point for research on self-organised construction.

2.1.4 Physical Embodiments

Swarm robots are simple robots that execute simple (reactive) behaviours. Their design is challenging because the designer is often assigned to a fixed budget which is the ultimate constraint of the swarm size. The lower the robot cost, the larger the robot swarm. That is why most research groups only operate 10s of robots [35][36] [37] or sometimes a few 100s [38][39]. Their cost range from tens of dollars [40] to thousands of dollars [35]. In the following, I highlight some of the swarm robots that have been designed over the past 25 years.

2.1.4.1 Khepera

Khepera depicts the first swarm robotic system that was built for the study of embodied swarm intelligence. It was introduced in 1993 and is illustrated in Figure 2.1. The Khepera swarm robot consists of three multi-layer PCBs stacked via PIN connectors. The bottom layer contains the sensory-motor board including two DC motors coupled with incremental sensors, eight analogue infrared proximity sensors and on-board power supply. A microcontroller controls the motors via pulse width modulation and four NMOS H-bridges. The infrared sensors are infrared LED and photo-transistor pairs. The microcontroller is mounted on the second layer. It is a 16MHz MC68331 Motorola microcontroller that controls all the peripherals of the robot. The third board holds two 64 pixel linear photo element arrays (cameras). There are other layers that can be stacked onto the Khepera. A manipulator turret makes Khepera capable of interacting with objects. The gripper is equipped with tactile sensors that give insights into the object size. The authors do not mention how much time it takes to build Khepera. It is interesting how similar today's swarm robots are to Khepera. In Section 2.1.4.4, I introduce GRITSBot, a swarm robot that was built in 2015, 22 years later. Khepera and GRITSBot both use differential drive, both increase functionality by stacking PCBs, and both possess a set of proximity sensors to detect and avoid obstacles. However,

the manufacture of robots has changed, where new manufacturing technologies such as pick-and-place machines make assembly significantly easier.

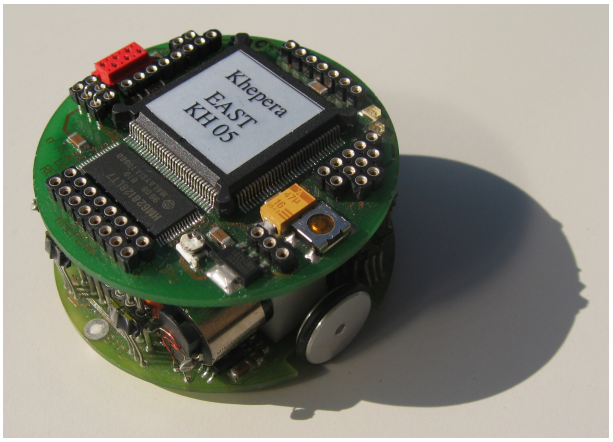


Figure 2.1: The first swarm robotic system: the Khepera robot (1993) [41].

2.1.4.2 E-puck

In 2009, 13 years after the development of Khepera, the developer of Khepera introduced another swarm robot: the e-puck [35]. The e-puck is one of the most sophisticated swarm robots that has ever been designed, however, it is also relatively expensive compared to other swarm robots. E-pucks possess eight infrared proximity sensors, a 3D accelerometer, three microphones, and a colour CMOS camera with a resolution of 640x480 pixels. E-pucks possess two motors and wheels using wheeled locomotion for movement. E-pucks use a speaker and an entire set (≥ 10) of red and green LEDs, that is distributed over the entire body chassis, as audio and visual outputs. They possess a bluetooth and Zigbee radio link. E-pucks come with a 64MHz dsPIC microcontroller which embeds a 16 bit processor and a digital signal processor (DSP) unit. E-pucks possess almost every piece of hardware that can possibly be embedded into a small robot. They can be purchased from *GCtronic* for $\$850$. That makes them difficult to operate in groups of 100s due to cost. That is why many swarm roboticists conduct research with far simpler and less expensive robot systems. The e-puck is an open-source robot.

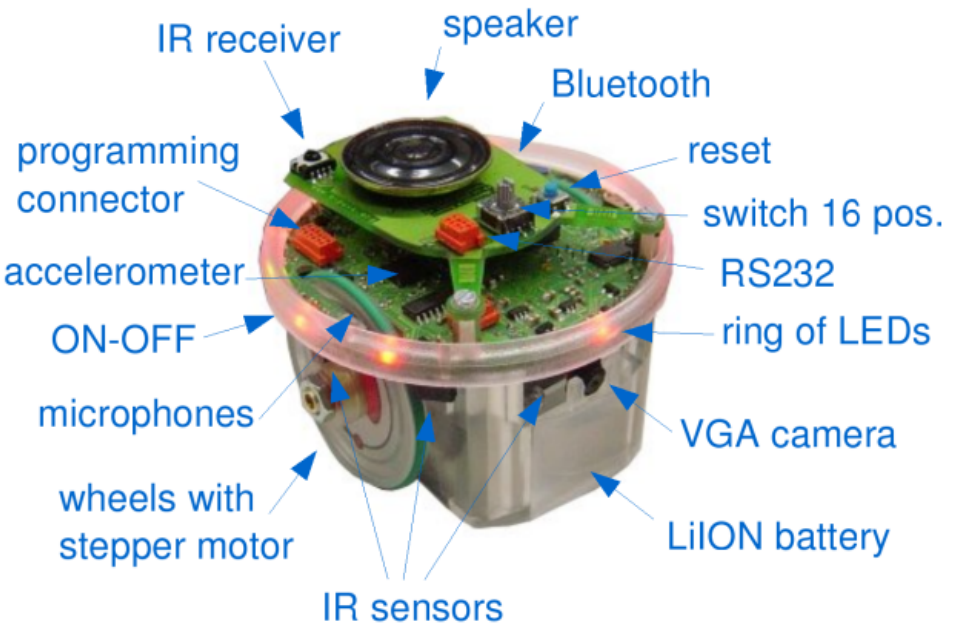


Figure 2.2: The e-puck is one of the most sophisticated and expensive swarm robots. It is equipped with a series of sensors, communication capabilities, and uses wheeled locomotion [41].

2.1.4.3 Kilobot

The Kilobot project has been an inspiration throughout my studies [40]. The robot was developed by Professor Mike Rubenstein at Harvard University. Kilobot exploits several unconventional design choices such as vibration motors for sliding movement and reflecting infrared light off the arena surface for communication and distance sensing between robots. Figure 2.3 shows Kilobot and a 2^{10} Kilobot swarm. The robot consists of a populated PCB, three legs, two vibrational motors, and a battery. It measures 33mm in diameter and 34mm in height. The robot uses slip-stick locomotion which I explain in Section 2.2. Robots communicate with one another via infrared. The communication range is approximately 3 robot distances. Kilobots are operated on a reflective arena surface for communication purposes. The infrared light is being reflected from the table surface back to adjacent robots. Kilobot is equipped with an ATMEGA 328 (8MHz) microcontroller. Kilobots possess a RGB light emitting diode (LED) and an ambient light sensor. Kilobots possess lithium ion batteries which supply robots up to ten hours with power. A charging spring allows a Kilobot swarm to be collectively charged. Kilobots cost $\$14$ in large quantities excluding assembly effort. In Section 2.2, I discuss

the assembly steps of Kilobots and compare them to other robots. The Kilobot swarm is the largest robot swarm that has ever been built and used in swarm robotic experiments [4]. However, Kilobots also possess a couple of disadvantages: 1) a single Kilobot requires five minutes of manual assembly; it takes a trained engineer approximately 85 hours to assemble a swarm of 1024 robots; 2) Kilobots' vibration motors have to be initially calibrated and eventually recalibrated. In general, the more attention an individual robot requires, the less operational becomes a large robot swarm.

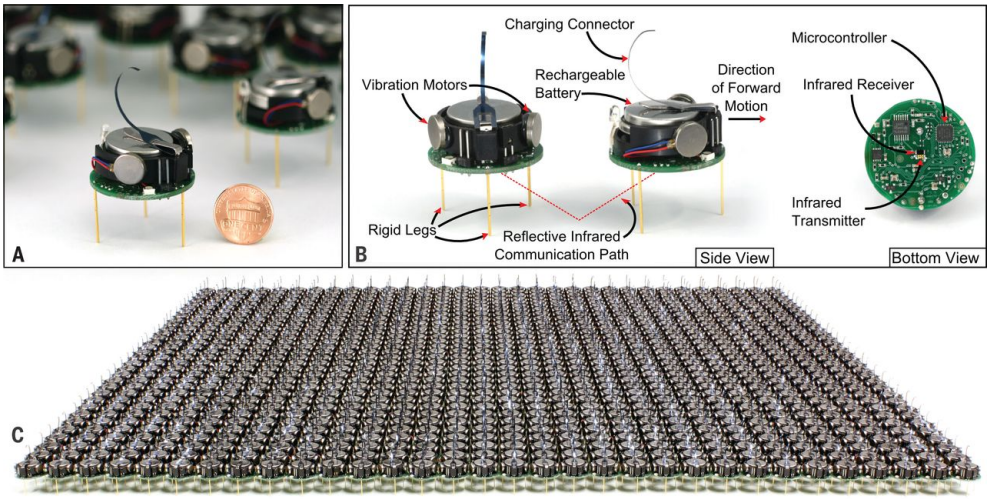


Figure 2.3: (A) A Kilobot robot, shown alongside a U.S. penny for scale. (B) Each Kilobot has an onboard microcontroller for executing programs autonomously, two vibration motors for moving straight or turning on a flat surface, and a downward-facing infrared transmitter and receiver. Robots communicate with others within a range of 10 cm (roughly three robot diameters) by reflecting infrared light off the table below. Communicating robots can evaluate relative distance by measuring the strength of the received infrared signal, but they cannot sense relative bearing (angle). (C) A 2^{10} Kilobot swarm. The Kilobot design allows for all operations on the entire swarm (charging, programming, etc.) to take a constant time to complete, independent of the number of robots in the swarm. Reprinted with permission from AAAS [4].

2.1.4.4 GRITSBot

Figure 2.4 shows GRITSBot, an inexpensive swarm robot [42]. It is used in the Robotarium, a remotely accessible multi-robot research facility, operated by Georgia Institute of Technology [43]. The facility allows students to program and run experiments with a swarm of GRITSBots. They offer support and tutorials on their webpage. This is an initiative to get students from all over the world involved in multi-robot research. GRITSBot is a differential drive swarm robot. It consists of three multi-layer PCBs that are stacked via PIN connectors, two motors, and two wheels. Some of GRITSBots components require manual assembly. I discuss GRITSBot's manufacture in Section 2.2. A GRITSBot costs $50 excluding assembly. Robots are equipped with ATMEGA 168 and ATMEGA 328 (8MHz) microcontrollers. The former microcontroller is solely used for motor control, the latter one for wireless communication, sensor data processing, and user-defined high-level tasks such as obstacle avoidance. A GRITSBot is manually assembled in one to two hours. A 1000 unit robot swarm of GRITSbots would require three months of non-stop manual assembly by trained engineers.

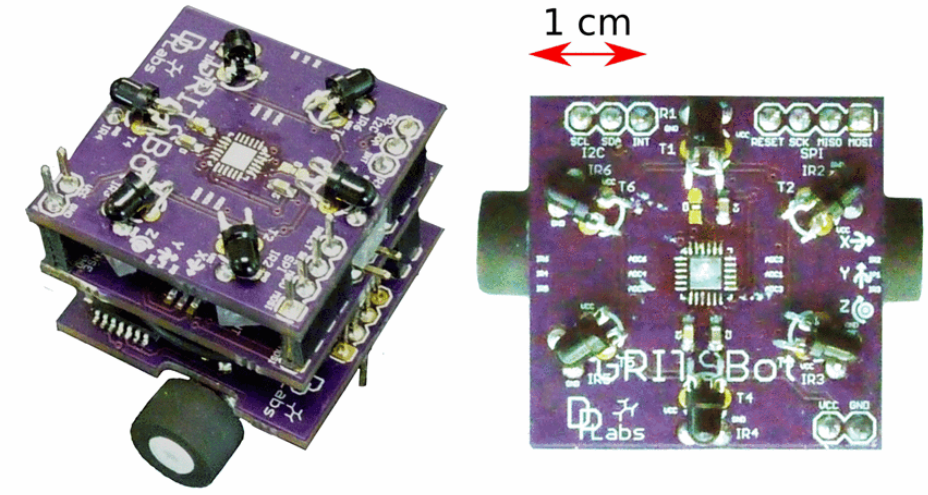


Figure 2.4: Isometric and top view of the GRITSBot [42].

2.1.4.5 Others

Figure 2.5 shows a collection of other swarm robotic systems. All robots but one use wheeled locomotion. They are equipped with a variety of sensors and communication capabilities. The S-bot is the only robot that possesses a gripper to interact with its environment and to build robot chains with other s-bots. In this collection, Droplet is the only robot that uses slip-stick locomotion, a locomotion strategy that I explain in Section 2.2. Slip-stick locomotion is low-cost to implement compared to wheeled locomotion, however, robots that use slip-stick locomotion are also generally slower. All robots possess infrared proximity sensors to detect and avoid obstacles.

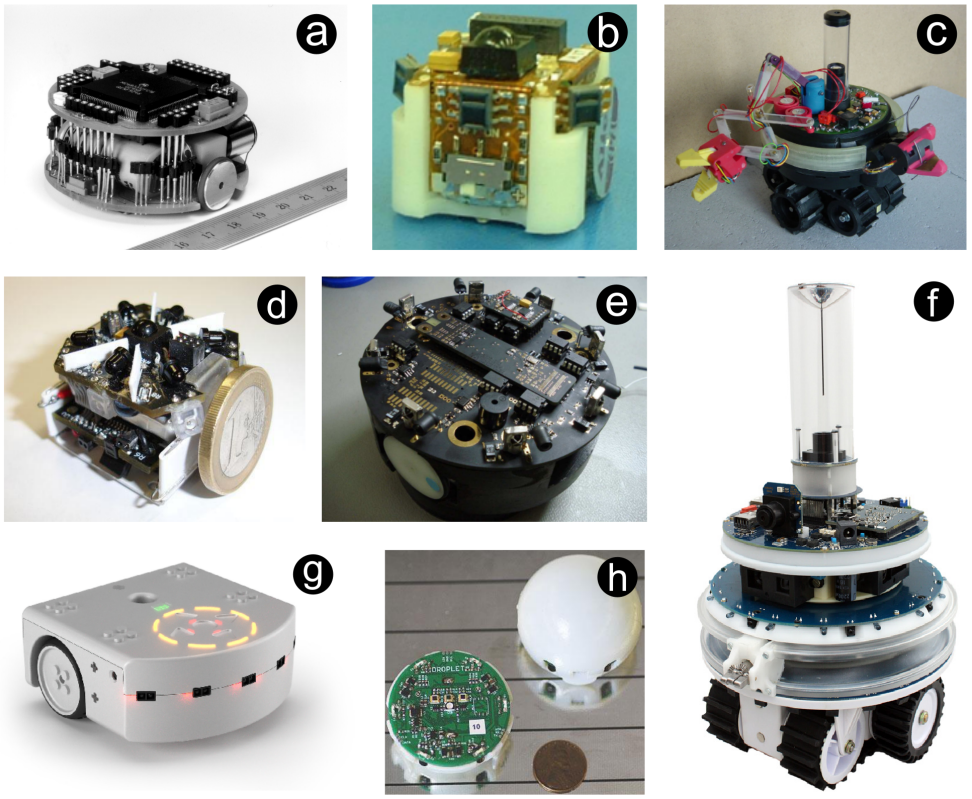


Figure 2.5: Collection of swarm robotics systems. A) Khepera [41], B) Alice [44], C) S-bot [45], D) Jasmine [46], E) Kobot [47], F) MarXbot [48], G) Thymio-II [49], and H) Droplet [50].

2.2 Locomotion Strategies of Swarm Robotic Systems

2.2.1 Robot Manufacture

The manufacture of a swarm robotic system is dependent on its design and components. Manufacture can consist of many assembly steps or just a few; the assembly steps can be easy or rather difficult, sometimes even requiring engineering expertise; and the time of assembly can range from several seconds to many hours or days. Mass-manufacture adds yet another facet: the total assembly time of a swarm robotic system is the time of assembly per single robot times swarm size. Therefore, swarm robots usually become these highly optimised designs, where often times an application bias determines the actual implementation. For example, if I want my robot swarm to collectively find a light source, I probably want them to posses light sensors.

It does not matter whether a component is part of the actuation or sensing circuitry. A component is a component. From a manufacturing perspective, I only care about the component's difficulty to be implemented. There are some electronics components for which assembly can be entirely outsourced, such as surface mount devices (SMDs). Such components can be assembled with pick-and-place machines eliminating the need for manual assembly. If a robot could only exist of a PCB and SMD components, its manufacture could be outsourced to 100%.

This has been one of the major achievements of my studies: the design and implementation of a PCB-only robot. I analysed which components of previous swarm robotic systems require non-SMD components. As it turns out, the locomotion strategy of a swarm robot is a major source of non-SMD components. Table 2.1 summarises the locomotion strategies of the most important swarm robotic systems and Table 2.2 compares the various locomotion strategies in more detail.

Ref.	Robot name	Locomotion strategy	Propulsion system	Hardware odometry
[41]	Khepera	Wheeled	DC motors	Wheel encoders
[44]	Alice	Wheeled	Bidirect. motors	Wheel encoders
[45]	S-bot	Treeled	DC motors	Wheel encoders
[46]	Jasmine	Treeled	DC motors	N/A
[51]	I-swarm	Slip-stick	Piezoelectric Polymer	No
[35]	e-puck	Wheeled	Stepper Motors	Wheel encoders
[47]	Kobot	Wheeled	DC motors	N/A
[48]	MarxBot	Treeled	Rotational motors	Accelerometer
[40]	**KiloBot**	**Slip-stick**	**Vibration motors**	**No**
[52]	R-one	Wheeled	DC motors	Wheel encoders
[49]	Thymio II	Wheeled	DC motors	N/A
[50]	Droplet	Slip-stick	Vibration motors	No
[53]	Elisa-3	Wheeled	DC motors	Wheel encoders
[42]	**GRITSBot**	**Wheeled**	**Stepper motor**	**Stepper motor**
[54]	Pheeno	Wheeled	DC motors	Wheel encoders
[5]	**HoverBot**	**Low-friction**	**Planar coils**	**Hall-effect sensor**

Table 2.1: Comparison of 16 swarm robotic systems found in the literature. The three highlighted rows depict the swarm robotic systems whose locomotion strategy is further analysed in Table 2.2.

Feature/Locomotion	**Wheeled** [42]	**Slip-stick** [40]	**Low-friction**
Robot velocity (cm/s)	Continous	Continous	Discrete - equidistant 20mm steps[1]
Battery lifetime	30min - 5h @ 150mAh	3 - 24h @ 160mAh	25min - 600h @ 300mAh
Dependencies	No	Flat surface	Magnet-levitation table
Hardware odometry	Stepper motors	No	Hall-effect sensor
Actuator calibration	Not required	Required	Not required
Number of non-surface-mount components[2]	≥ 4	≥ 5	0
Difficulty of mechanical assembly[3]	(1)(2)(3) (4)(5)(6)	(1)(3)(6)	(6)
Cost[4] **at 1,000 units ($)**	13.34	3.12	1.96

Table 2.2: Comparison of wheeled, slip-stick, and low-friction locomotion.

[1]Robust (error tolerant) movement on a discrete grid is equivalent to precise movement.

[2]Wheeled: two wheels, two motors, and motor control board. Slip-stick: three legs, two vibration motors, and electronics. Active low-friction: electronics.

[3](1) Soldering non-surface-mount components, (2) cutting components, (3) gluing components, (4) screwing components, (5) stacking components, and (6) connecting battery.

2.2.2 Wheeled and Slip-Stick Locomotion

Tables 2.1 and 2.2 contain specific terminology. While most terminology for these features is self-evident, I provide a summary here for those that may be unclear. *Hardware odometry* refers to the use of sensors to estimate change in position over time. This term indicates systems which do not possess a real form of odometry or which address the lack of hardware odometry by performing collective algorithms [40]. In this column, N/A refers to the fact that the cited publication does not explicitly state information about odometry. *Type of motion* clarifies whether a motion is continuous or discrete and if discrete with what step size. *Dependencies* refer to specific environments which the robots require to function properly. *Surface-mount-technology components* are components which can be directly soldered onto a PCB. *Non-SMD components* are usually incompatible with pick-and-place machines and often require manual assembly which generally increases the labour effort and cost of mass-manufacture.

The swarm robotic systems listed in Table 2.1 use either wheeled or slip-stick locomotion. Slip-stick locomotion refers to the alternation between slipping and sticking of an agent to a substrate that results into directed locomotion [55]. The vast majority of swarm robotic systems use wheeled locomotion with DC motors and wheel encoders. There are a few exceptions which use tracks and wheels (treels) and accelerometers, gyroscopes, or stepper motors for odometry. Treels are considered as wheeled locomotion. Three systems use slip-stick locomotion, whereas two of those three systems use vibration motors, and the remaining system uses piezoelectric polymers as actuators. The HoverBot system, presented in this thesis, implements low-friction locomotion.

Table 2.2 compares wheeled, slip-stick, and low-friction locomotion by using the GRITSBot, the Kilobot, and the HoverBot as representative systems. I select Kilobot as a representative for slip-stick locomotion because it is the first and only large-scale robot swarm exceeding a collective size of 1,000 units. I choose GRITSBot as a representative for wheeled locomotion. Pickem et al. present in [42] a recent system that explores both cost and functionality of wheeled robots.

[4]Cost of components that are solely associated with locomotion, in order quantities of 1,000.

While wheeled locomotion has advantages in robot velocity, platform independence, hardware odometry, and actuator calibration, it has disadvantages in battery lifetime, number of non-SMD components (minimum two wheels and two motors), difficulty of mechanical assembly, and cost (including motor control board). In Pickem et al.'s work, non-surface-mount components had to be soldered, receiver coil wires needed to be cut and glued, wheels had to be screwed onto motors, circuit boards needed to be stacked, and the battery had to be connected.

In comparison, slip-stick locomotion has advantages in battery lifetime and cost, but disadvantages in robot velocity, the dependency on flat surfaces, hardware odometry, actuator calibration, and number of non-SMD components (the minimum number being three legs and two vibration motors). In Rubenstein et al.'s work, their mechanical assembly consisted of soldering non-surface-mount components, gluing vibration motors to the robot, and connecting a battery.

Although the GRITSBot possesses more functionality than the Kilobot, Kilobot's locomotion strategy favours mass-manufacture. The low manufacturing effort of slip-stick locomotion is one of the reasons why Kilobot has been manufactured in large quantities. Not only is the assembly of robots that use slip-stick locomotion faster, it is also less difficult to maintain a robot that is built from fewer parts and assembled with less instructions.

Low-friction locomotion has advantages in that it provides hardware odometry, requires no actuator calibration, has no non-SMD components, simple mechanical assembly, and is low cost; but it has disadvantages in robot velocity, dependency on a magnet-levitation table, and battery lifetime. The mechanical assembly of HoverBot consists of plugging in a battery to the robot. I describe the HoverBot system in Chapter 3.

Overall, each of the three strategies possesses certain advantages over the others.

Not included in Table 2.1, but relevant to my technical approach, is work from Vladimerou et al. (2004), Gross et al. (2011), Napp et al. (2011), Cappelleri et al. (2014), and Pelrine et al. (2017). Vladimerou et al. developed a group of autonomous hovering robots. Robots were miniaturised air cushion vehicles (ACVs) also called hovercrafts. Such robots use blowers to create air cushions beneath their contact surfaces and propel by utilising additional air

blowers. Gross et al. reported on an experimental setup in which they investigated aided assembly with floating building blocks using an air table [56]. Their system used low-friction locomotion in which their building blocks did not possess locomotion capabilities, but modules would flow passively in the agitated medium. Napp et al. investigated stochastic interactions between active and passive robots using low-friction locomotion [57]. Passive robots were foam blocks with complementary shape and embedded magnets that assembled over time on an air bed. Active robots, while not capable of autonomous movement, could expend energy to disassemble the passive robots. Cappelleri et al. introduced a novel approach to achieving independent control of multiple robot magnets [58]. In their work, they designed a grid of planar microcoils. The coils were used to generate magnetic potentials to control the trajectories of magnets. This approach produces the simplest swarm robots that currently exist: permanent magnets. Pelrine et al.'s work is similar to Cappelleri's but differs in that they add a thin graphite layer onto their PCBs which makes their magnet robots levitate [59]. Both of these publications feed into additive micromanufacturing with swarms. Similarly to Gross's and Napp's work, agents did not possess locomotive autonomy but were moved by external stimuli. All five approaches are relevant but distinctly different to the work I present here.

2.3 Sensing Capabilities of Swarm Robotic Systems

The sensing capabilities of a robot influence the type of experiments one can perform. While a camera adds more functionality to a robot than an ambient light sensor, each sensor comes at a different cost. I systematically analysed previous swarm robotic systems and found that some systems possess sensors that have been, or could be, used for the detection of multiple signals. For example, an IR transceiver could be used for communication and proximity sensing amongst others. Developing robots that are low-cost and functional is a challenging task, therefore, utilising sensors for the detection of multiple measurands is desirable.

In this section, I introduce an instrument model, a well-established model borrowed from the sensors community to generally describe a measuring device, to establish a clear understanding of sensors and how they can become multi-functional. Then I give a comprehensive review on the sensing capabilities of previous swarm robotic systems and categorise them based on their multi-functionality.

2.3.1 The Instrument Model

The instrument model shown in Figure 2.6 is a scientifically accepted model from the sensor community to generally describe a measuring device [60]. An instrument is a device that transforms a physical variable of interest — the measurand — into a form that is suitable for recording, the measurement, as conceptually shown in Figure 2.6. An example of a basic instrument is a ruler. In this case the measurand is the length of some object and the measurement is the number of units (meters, inches, etc.) that represent the length.

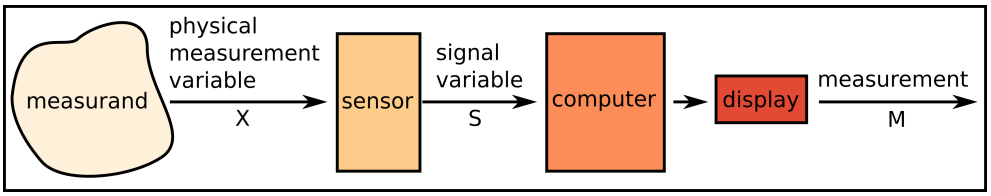


Figure 2.6: Instrument model [60]. The measurand is the measured value of interest, whereas the physical measurement variable is associated, either directly or indirectly, to the measurand. The sensor converts the physical measurement variable in a signal variable (often an electric signal), and feeds it into a processing unit or computer. The value that we actually display is the measurement.

Any measurand (distance, collision, temperature, etc.) is linked to an observable physical measurement variable X. The observable physical measurement variable X does not necessarily have to be the measurand, X can only be related to the measurand. For instance, the mass of an object is often measured by the process of weighing, where the measurand is the mass but the physical measurement variable is the downward force the mass exerts in the Earth's gravitational field. Collision detection is another example. A robot can detect the measurand collision by measuring force or by relating the measurand to another physical measurement variable such as acceleration. In this case you can either purchase an accelerometer or a set of force sensors (e.g. 4 force sensors - one sensor on each robot side). Both implementations allow the detection of collisions, however, the single accelerometer is likely going to be cheaper than the force sensors. There are many more of such examples, but there are also variants in which a single physical measurement variable contains information about several measurands. An excellent example is infrared

light. Infrared light can be used for the measurement of distance, to determine bearing and to communicate with other robots [50]. I call this capability *multi-functional sensing*. Communication is usually handled by a transceiver; you transmit and receive or transceive data by means of a physical channel e.g. by utilising electromagnetic waves in the infrared spectrum. The receiving of signals requires sensors, such as photodiodes that transduce infrared light into electric signals, hence communication itself can be considered as a sensing task.

2.3.2 Multi-Functional Sensing

I reviewed the sensing capabilities of 15 swarm robotic systems found in the literature and I summarised the findings in my journal publication [5] (Appendix F). Table 2.3 is a subset of Appendix F containing robot systems capable of multi-functional sensing with ≥ 2 measurands. The content of Table 2.3 is based on the cited work shown in the first column of each row. Table 2.4 takes a closer look at the few robot systems capable of multi-functional sensing with ≥ 3 measurands.

Robot System / Number of Measurands	2	**3**	**4**
Khepera (1994) [41]	IR		
Alice (2003) [44]	IR		
SBot (2003) [45]	L		
Jasmine (2005) [46]	IR		
E-puck (2009) [35]	IR, L		**AC**
MarXbot (2010) [48]	IR, L		
Kilobot (2012) [40]	IR		
R-One (2013) [52]	IR		
Droplet (2014) [50]		**IR**	
GRITSBot (2015)[42]	IR		
Pheeno (2016) [54]	AC		
HoverBot (2017) [5]		**MF**	

Table 2.3: Analysis measurands from previous swarm robotic systems. IR: Infrared Light, L: Visible Light, AC: Acceleration, MF: Magnetic Field. Other systems that I considered but ended up using single-measurand sensors are: SwarmBot [52], Kobot [47], and Thymio-II [49].

Measurand / Physical Measurement Variable	**Droplet** [50] **(infrared)**	**E-puck** [35] **(acceleration)**	**HoverBot** [5] **(magnetic field)**
Local communication	x		
Distance	x		
Bearing	x		
Inclination		x	
Collision		x	x
Free-fall		x	
Movement		x	
Odometry			x
Rotation			x

Table 2.4: Further comparison of swarm robotic systems with 3 or more measurands per sensor.

The majority of swarm robotic systems (12/15) are capable of multi-functional sensing with ≥ 2 measurands. In the ≥ 2 measurand category, the most commonly used physical measurement variables are IR light, followed by ambient light. IR light has been mainly utilised for distance/proximity sensing and local communication, whereas ambient light has been often utilised for object detection and long-range distance measurements through a camera.

To the best of my knowledge, there are only two swarm robotic systems that are capable of, or make use of, multi-functional sensing with ≥ 3 measurands. The e-puck is capable of measuring four measurands with an accelerometer [35]. It measures inclination, collision, free-fall and movement acceleration. The Droplet is capable of measuring three measurands with a set of IR sensors [50]. It uses six symmetrically placed IR sensors to measure distance, bearing and local communication. Therefore, once a robot possesses an accelerometer and a group of IR sensors, it is capable of measuring seven measurands by only using two different types of sensors. HoverBot is capable of measuring three measurands with a single Hall-effect sensor [5]. It associates magnetic field readings to collision, successful movement, and rotation events. The Hall-effect sensor was integrated into HoverBot's circuitry because HoverBots are exposed to a magnetic grid that is embedded into the top surface of their arena.

Figure 2.7 summarises the multi-functional sensing capabilities of 15 swarm robotic systems. The vast majority of swarm robotic systems only use their sensors to measure a single modality or two modalities. There are other sensors

that provide more capability (measuring ≥ 2 modalities) such as accelerometers. In general, choosing sensors for swarm robots should be regarded as functionality-cost optimisation in which swarm roboticists embed those sensors that maximise the overall robot capability for a given budget.

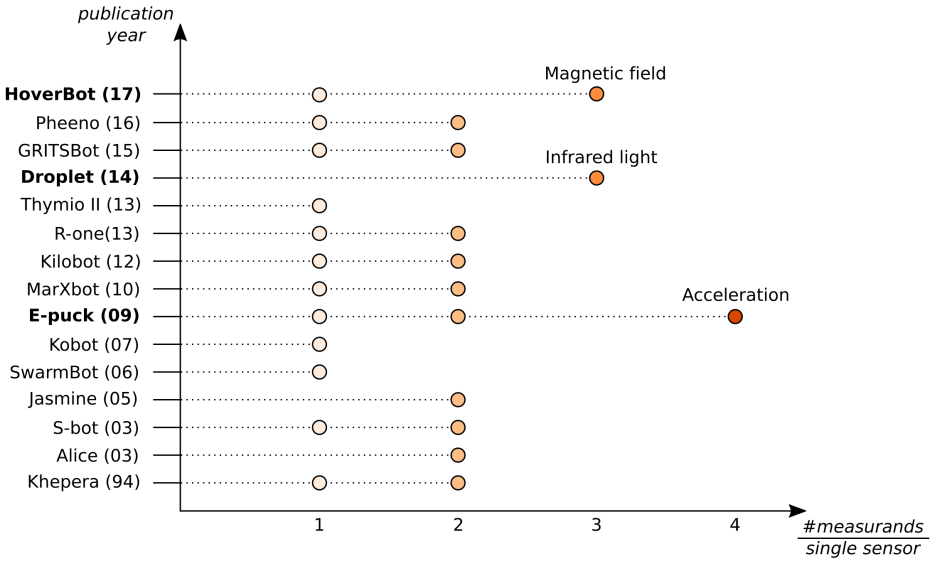


Figure 2.7: Overview of the sensing capabilities of previous swarm robotic systems. Swarm robots are sequentially listed according to their publication date. The x-axis indicates how many measurands can be measured per single sensor. Although most swarm robotic systems utilise their sensors to detect up to two measurands, there are only three robot systems including the HoverBot that utilise their sensors to measure 3 or more measurands.

The instrument model comes with its limitations: the measurand component can be interpreted from different angles. For example, successful movement and rotation can be generally considered as being part of a robot's odometry capability. However, the to-be-measured value, the measurand, is not odometry but successful movement and rotation. Since the instrument model is to some extent subjective, it is of paramount importance to apply this model consistently. Figure 2.7 is a collection of carefully categorised sensing capabilities of swarm robotic systems. A key understanding that I have derived from studying the instrument model is that, in swarm robotic systems, often the sensors could be further utilised, and therefore the systems should be reanalysed.

2.4 Physical Interactions in Swarm Robotics

In large groups of robots, by necessity, collision avoidance becomes the dominant behaviour as the robot density increases [61]. Robots are busy avoiding each other rather than performing the task at hand. The vast majority of (swarm) robotic systems avoid collisions. Collisions are perceived as worst case scenarios and an experimental byproduct that often times is avoided at all cost. In such cases, robots are equipped with specific hardware that allows them to detect obstacles and avoid collisions. Collision avoidance is a high priority behaviour that overrules most other robot processes.

Embracing and detecting collisions - workshop video

2.4.1 Collision Avoidance in Robotics

The vast majority of swarm robotic systems are wheeled robots, which I discuss in Section 2.2. Wheeled robots are usually made of hard materials, are heavy, and reach high velocities creating large momentum ($\vec{P} = m\vec{V}$), which can lead to destructive collisions. Therefore most wheeled robot systems avoid collisions to keep the robot and its immediate environment safe [62]. Collision avoidance becomes an integral part of the robot design; resources are spent on sensors and low-level control schemes. Most robots detect obstacles through some kind of contactless sensor such as optical (infrared, laser ranging, etc.) or sonar sensors. In addition, it requires computational resources to analyse the raw sensor data. This indicates that collision avoidance has an impact on robot hardware. If I want to build a simple robot that is required to avoid collisions, a large percentage of my budget goes to collision avoidance. Moreover, each component on a robot has an impact on the robot's manufacturability as I discuss in Section 2.2. Overall, collision avoidance is a robot skill that makes a robot more expensive. Swarm robotics is a research field that is heavily inspired by colliding species. Many natural swarm systems such as bees do not avoid collisions at all cost, they embrace and/or are influenced by physical interactions [63]. Therefore, I belief that there is an increasingly growing narrative for research on collision-based collective behaviours.

2.4.2 Collisions in Biological Systems

In biological swarm systems, collision avoidance seems to be a question of agent size. While large biological agents often avoid each other, such as fish [64] or birds [65], small biological agents tend to physically interact:

If a bee knows about a suitable nest site, it performs a waggle dance to communicate its discovery to its peers. If another bee knows about a good nest site too, it tries to stop the waggle dance by pushing its head against the dancing bee [66]. Physical interaction becomes a feedback mechanism that is paramount for communication and decision making in large bee collectives. Japanese honeybees (*Apsis cerana japonica*) fight against giant hornets by forming a hot defensive bee ball. They surround the hornet and the physical interactions between large groups of bees produce heat, killing the hornet [67]. Ants (*Temnothorax albipennis*) use a technique known as tandem running to lead each other from nest to food source by using physical interactions such as tapping legs and abdomen with their antennae to control the speed and course of their run [31]. Ants (*Solenopsis invicta*) use physical interactions to survive in water. They link their bodies together to build waterproof rafts that enhance their water repellency for survival [68]. Some forms of cell migration emerge from physical interactions (inelastic collisions) between cells [69]. This phenomenon plays a fundamental role in tissue growth, wound healing, and immune response. A study on granular media makes comparisons to biologically inspired interacting agents and shows that simple inelastic collisions between self-propelled agents can provide a wide range of self-organised collective behaviours [70].

There are many reports on collision-triggered behaviours in nature, however, there have been only a few swarm robotic studies that concern collisions.

2.4.3 Collision Dependent Robot Behaviours

There are only a few swarm robotic studies that deal with collisions. Kernbach et al. and Schmickl et al. worked on the re-embodiment of biological aggregation behaviours of honeybees. They show how to take advantage of collisions to develop scalable robot behaviours. In their work, swarm robots converge to light sources without requiring inter-robot communication. Concretely, they minimize sensing and computation by evaluating robot data only once per collision; more frequent collisions lead to more data evaluations [39][71]. Mayaa et al. harnessed collisions to help localise a robot within an arena. The arena was divided into differently sized segments, whereas each segment was inhabited by differently sized robot groups. Robots used collision detection as an information source to determine their locations [61].

Overall, collision is a promising candidate for research on and the design of scalable robot behaviours, since the occurrences of collisions usually increase with increasing group sizes. Scalable refers to the ability of a swarm to perform well with different group sizes; the introduction or removal of individuals does not result in a drastic change in the performance of a swarm (see Section 2.1.2).

2.5 Summary

The field of swarm intelligence is a biologically driven research field. It originates from observations in nature. Examples range from insect colonies to the immune response. Large collectives of simple agents accomplish complex tasks through simple interactions. Such collective systems exhibit properties (robustness, flexibility, and scalability) that are very useful for robotics applications. Swarm systems can perform a variety of behaviours, whereas applications are compositions of behaviours. In the past 25 years, researchers have developed many (10s) swarm robotic systems, whereas most of them are wheeled robots that use differential drive. The locomotion strategy of a robot has a major impact on the robot's manufacture. Wheeled robots are usually more difficult to manufacture than robots that use slip-stick or low-friction locomotion. The sensing capabilities of a robot impact its functionality. However, some sensors can be used for a variety of tasks (e.g. infrared sensors for communication, distance, and bearing sensing), whereas other sensors are very specialised (e.g. humidity sensors). The instrument model from the sensing and measurement community helps analyse sensors for their multi-functionality. Accelerometers are the most multi-functional sensors that have been embedded into swarm robotic systems. Researchers have detected inclination, collision, free-fall, and movement with accelerometers. Swarm robotic systems usually avoid physical collisions. Collisions are perceived as worst-case-scenarios that often have to be avoided at all cost. However, the ability to detect collisions makes robots more expensive minimising the budget for other robot functionalities. In biological swarm systems, collision avoidance seems to be a question of agent size. While large biological agents often avoid each other, small agents tend to physically interact. There have been only three studies on collisions with swarm robotic systems. In these studies, robots did not collide, but approximated collisions with distance sensors.

Chapter 3

The HoverBot System

The HoverBot system is an easily manufacturable swarm robotic platform that uses low-friction locomotion. It consists of a magnet-levitation table and hovering robots called HoverBots. Figure 3.1 exemplarily shows the robots and Figure 3.2 the HoverBot system. In this Chapter, I discuss the guiding principles, permutations, and details of the HoverBot system. This chapter is based on my published journal paper [5] which I put in Appendix E and a patent application filed by the University Court of the University of Edinburgh [6].

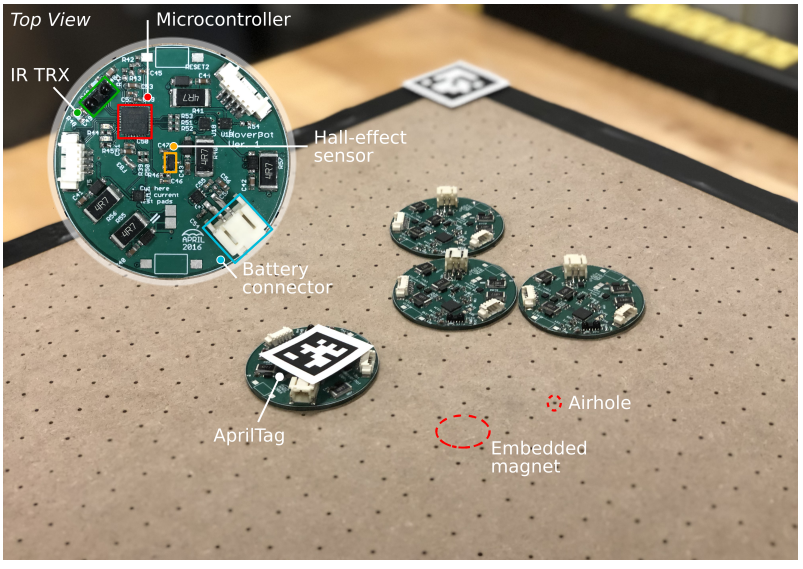


Figure 3.1: Introduction to HoverBots. HoverBots are floating circuit boards that operate on a specialised arena: the magnet-levitation table.

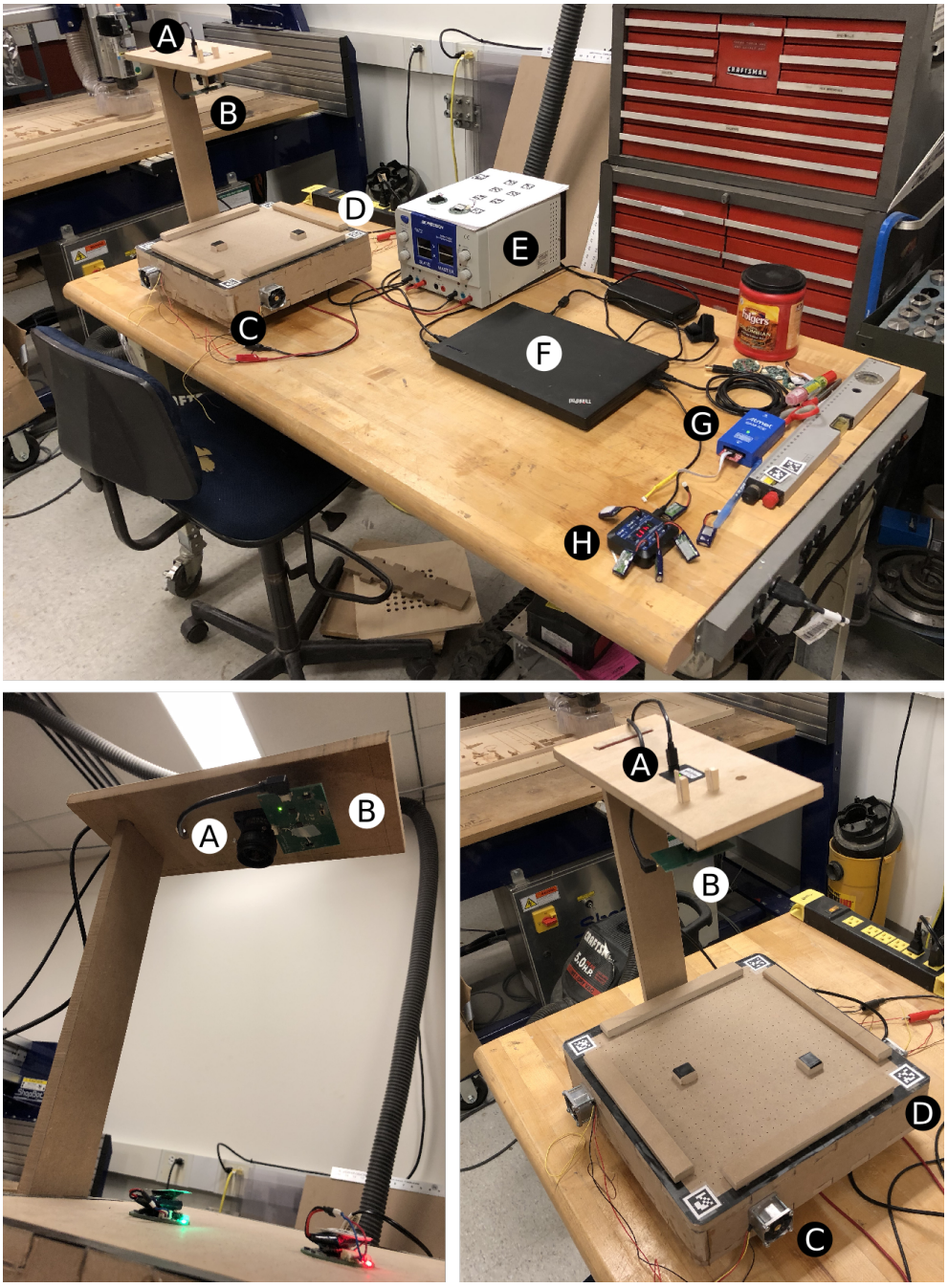


Figure 3.2: Assembly of the entire HoverBot system. The HoverBot system consists of: A) camera system; B) infrared communication system; C) four fans; D) AprilTags; E) auto-transformer; F) computer; G) programmer; and H) battery charger.

3.1 Low-Friction Locomotion

To move — on land, in water, or in the air — always requires an expenditure of energy. Reducing the resistance to motion, namely, friction, allows a greater range of travel for a given input of energy [72]. Low-friction locomotion is a class of locomotion strategies that focuses on reducing friction between robots and their substrates to facilitate robot movement.

3.1.1 Free Body Diagram of a Robot

Figure 3.3 shows the free body diagram of a robot. According to Newton's first law of motion, the robot remains still unless acted upon by unbalanced forces. In this simplified model, there are four forces that act on a robot: 1) gravitational force, 2) normal force, 3) frictional resistance (static and kinetic), and 4) propulsional force.

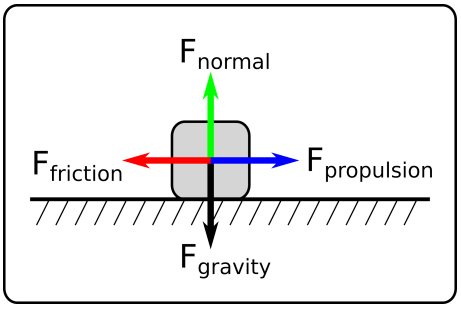


Figure 3.3: Free body diagram of a robot. The net force vector determines the movement direction of the robot.

The gravitational force is dependent of the mass of the robot.

$$\vec{F}_{gravity} = M_{robot} \times \vec{g} \tag{3.1}$$

Assuming no other force opposes the gravitational force, the normal force is in magnitude equivalent to the gravitational force.

$$|\vec{F}_{normal}| = |\vec{F}_{gravity}| \tag{3.2}$$

A robot has to overcome static friction before it can move. Forces smaller than $F_{static,friction,max}$ are opposed by frictional resistance of equal magnitude

and opposite direction. Once static friction has been overcome, static friction does not apply anymore.

$$\vec{F}_{static,friction,max} = \vec{F}_{normal} \times \mu_{static} \tag{3.3}$$

Kinetic friction occurs when a robot starts moving. It has to continuously overcome kinetic friction to keep moving. The coefficient of kinetic friction is usually lower than that of static friction.

$$\vec{F}_{kinetic,friction,max} = \vec{F}_{normal} \times \mu_{kinetic} \tag{3.4}$$

The net force is the sum of the individual forces:

$$\vec{F}_{net} = \vec{F}_{gravity} + \vec{F}_{normal} + \vec{F}_{friction} + \vec{F}_{propulsion}. \tag{3.5}$$

Any propulsion force that exceeds the frictional resistance unbalances the net force and causes motion.

3.1.2 Reducing Frictional Resistance

Since propulsion force and frictional resistance are opposing forces, the minimum propulsive force that is required for motion can be lowered by reducing the frictional resistance in the first place. The frictional resistance can be either partly reduced or completely eliminated as shown in Figure 3.4.

Equation 3.6 indicates that the frictional resistance can be either *reduced* by decreasing the mass of a robot or by using materials with low friction coefficients.

$$\vec{F}_{friction} = M_{robot} \times \vec{g} \times \mu_{material} \tag{3.6}$$

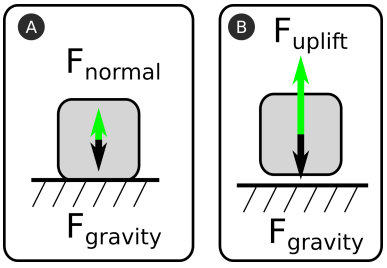


Figure 3.4: A) The frictional resistance can be countered for example through a decrease of robot weight or B) by an uplifting force that exceeds the robot's gravitational force. The normal force is of equal magnitude and opposite direction to the gravitational force if a robot remains still and makes contact with a surface.

The frictional resistance can be completely *eliminated* by inducing an uplifting force equivalent to or larger than the gravitational force of the robot also called levitation. In this case, the robot does not make contact with the arena surface anymore; the normal force is substituted by an uplifting force which eliminates the frictional resistance. Examples of uplifting forces include: i) electro-static levitation in which an electric field is used to levitate a charged object [73]; ii) magnetic levitation in which magnetic forces are used to uplift other magnetic objects [74]; iii) aerodynamic levitation in which a stream of gas is used to make objects float [75]; iv) acoustic levitation in which acoustic waves are being used to uplift objects [76]; v) optical levitation in which a laser uplifts objects via photon momentum transfer [77]; or vi) buoyant levitation in which pressurised gases possess high densities and lift objects from the ground (e.g. helium balloon).

Table 3.1 shows the coefficients of friction for materials ranging from rubber to Teflon. Friction coefficients are always dependent of two materials. Table 3.1 helps identify materials that reduce the frictional resistance. Both the materials of the arena and robot, determine the frictional resistance of the system. Any material with low friction coefficient reduces the frictional resistance against motion.

Material	**Kinetic**	**Static**
Rubber on concrete (dry)	0.68	0.9
Rubber on concrete (wet)	0.58	-
Rubber on asphalt (dry)	0.67	0.85
Rubber on asphalt (wet)	0.53	-
Rubber on ice	0.15	-
Waxed ski on snow	0.05	0.14
Wood on wood	0.3	0.42
Steel on steel	0.57	0.74
Teflon on teflon	0.04	-

Table 3.1: Coefficients of friction for a variety of different materials [78]. These are typical examples of coefficients of friction. The coefficients are subject to changes due to material compositions.

If the frictional resistance has been reduced or eliminated, additional forces can be used to propel the robot. In Section 2.2.2, I discuss several robot systems that utilise external propulsion forces for locomotion, that is, the robot is being moved heteronomously rather than autonomously. In general, the lower the frictional resistance, the easier it is i) to move the robot or ii) for the robot to move. As illustrated in Figure 3.5, eliminating frictional resistance by encountering gravitational force with an uplifting force minimizes the absolute propulsive force that is required for motion.

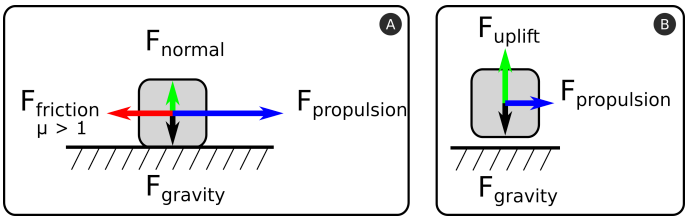


Figure 3.5: A) The magnitude of the propulsive force for movement is dependent on the frictional resistance. B) If the gravitational force is encountered through an uplifting force, the propulsive forces that are required for movement become minimal.

3.1.3 Implementation

There are different ways of implementing low-friction locomotion, the HoverBot system implements it through a specialised robot arena: the magnet-levitation table which is conceptually illustrated in Figure 3.6. A stream of air lifts robots from the arena surface. Robots generate propulsive forces by interacting through their electromagnetic anchors with magnets that are embedded into the arena surface.

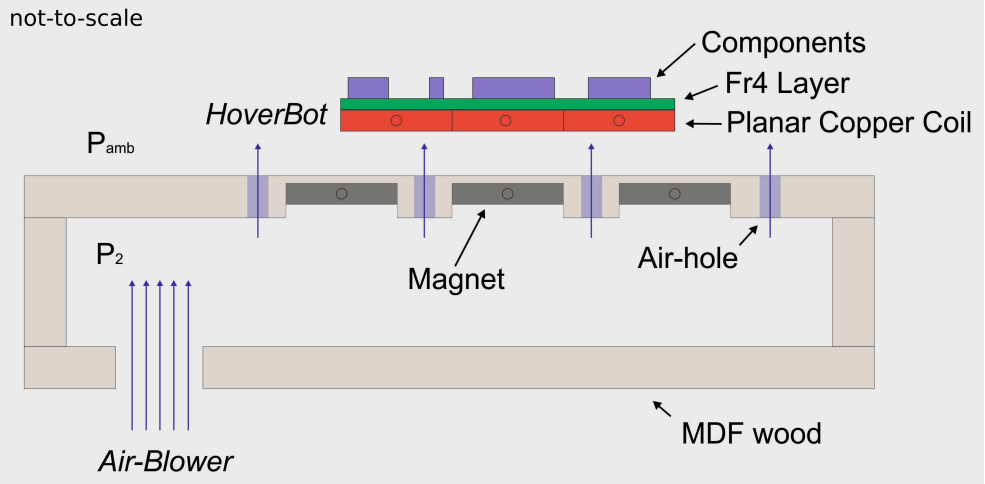


Figure 3.6: Conceptual figure: the dimensions of objects are not proportional. The HoverBot system implements low-friction locomotion through a magnet-levitation table. An upstream of air lifts robots from the arena surface; robots interact through their electro-magnetic actuators with permanent magnets that are embedded into the arena. Appendix C contains the technical drawings of the magnet-levitation table.

An air source blows a mixture of gases, namely nitrogen, oxygen, argon, and carbon dioxide, also known as dry air, into an air chamber. Air molecules are pushed into the chamber with a constant flow rate. At first, more air molecules enter the chamber than escape. After a while, the pressure in the chamber is greater than the ambient pressure, a constant differential pressure equilibrates, and the input and output airflow rates become equal. Although the air motion and the air pressure distribution in the thin gap between agent and tabletop may be quite complicated, they can be neglected as long as agents are able to float anywhere on the magnet-levitation table.

A good example of HoverBot's locomotion can be observed in nature. *Nannosquilla decemspinosa* is a small stomatopod found in sand substrates on the Pacific coast of Central and South America and illustrated in Figure 3.7. These stomatopods are capable of maneuvering if supported by a 1-mm layer of water and lose this capability once their surrounding dries up [79]. The water reduces the stomatopod's frictional resistance to the ocean floor; the weak propulsive forces from their legs are sufficient for locomotion.

Figure 3.7: Stomatopods swim if there is as little as a millimetre of water. If the sand is well drained and only moist, they are incapable of using their legs and change to a different form of locomotion. The dimensions of the stomatopod illustrated in this figure are unknown, however, mantis shrimps usually grow to around 10 centimetres. Image credit: National Science Foundation Multimedia Gallery [80].

3.2 The Magnet-Levitation Table

The magnet-levitation table is a key component of the HoverBot system. It supplies robots with an airflow and static magnetic fields. Figure 3.8 shows the magnet-levitation table from patent application GB1611448.0 filed by the University Court of the University of Edinburgh.

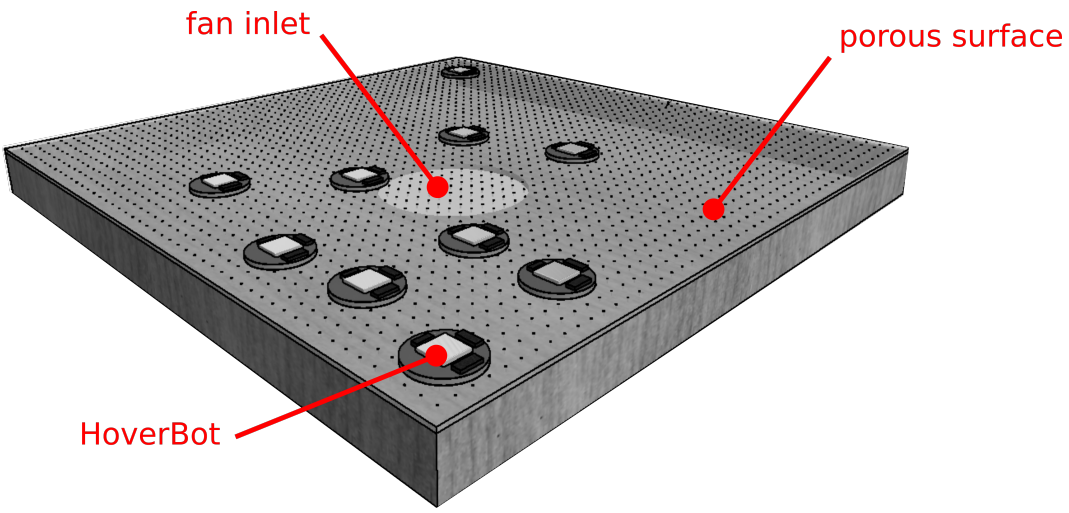


Figure 3.8: Illustration of the magnet levitation table from patent application GB1611448.0.

3.2.1 Iteration History

Figure 3.9 shows the different magnet-levitation table iterations that I have fabricated during the course of my studies. The first and second table iteration consisted of acrylic sheet and magnet holes were cut with a laser cutter. The third and final iteration is made of Medium Density Fibre (MDF) board and magnet pockets are milled with a Computer Numerical Control (CNC) machine. Magnets are hidden $1\mathrm{mm}$ beneath the table surface. The first and second table iterations used a leaf-blower as air-source. The final table iteration has four high-pressure fans mounted on its side-walls.

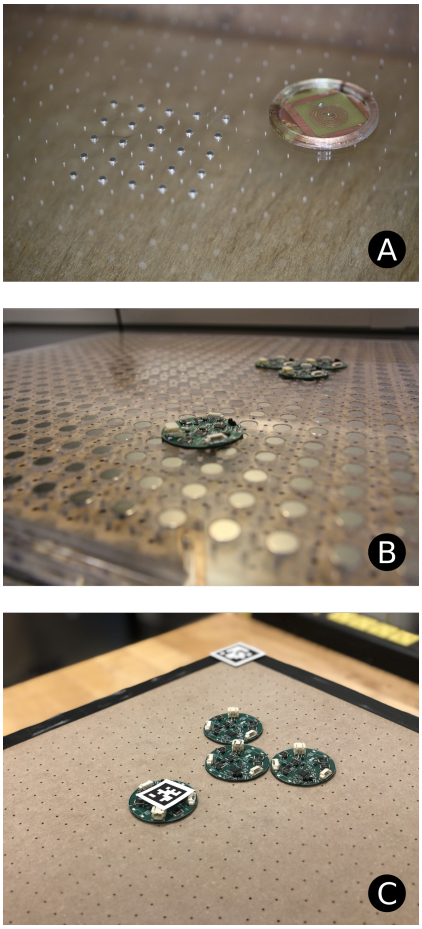


Figure 3.9: History of the magnet-levitation tables. A) Thin acrylic sheet with small permanent magnets embedded into top surface. B) Thick acrylic sheet with magnets that match robot coil diameter embedded into top surface. C) MDF board with magnets hidden beneath top surface.

3.2.2 Fluidic Mechanic Calculations

Figure 3.10 shows an overview of the fluid mechanics of the magnet-levitation table. P_R is the pressure within the table's reservoir, P_a is the ambient pressure, M is the mass of the robot, and R is the radius of the robot defining its size.

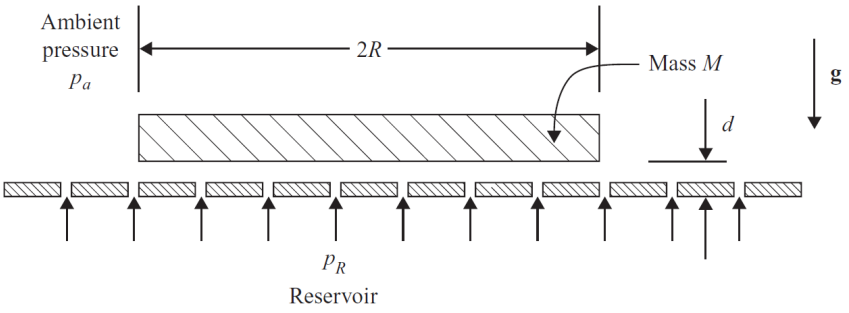


Figure 3.10: Fluid mechanics of the magnet-levitation table. An object of mass M and diameter $2 \times R$ levitates above a substrate. The differential pressure between ambient pressure P_a and reservoir pressure P_R has to be at least as high as the pressure evoked by the object to lift it from the substrate. Image credit and permission: Cambridge University Press, Appendix D.1.

3.2.2.1 Robot Payload

The weight and size of an agent determines the differential pressure that is required for levitation as I indicate in Equation 3.7.

$$P_{agent} = \frac{F_{agent}}{A_{agent}} = \frac{M_{agent} * g}{\pi * R_{agent}^2} \tag{3.7}$$

P_{agent} is proportional to R_{agent}^{-2} and M_{agent}.

3.2.2.2 Differential Pressure

The minimum reservoir pressure that could possibly produce sufficient force to lift the robot from the table is

$$\Delta P = (P_R^{'})_{min} - (P_a^{'}) \geq \frac{M \times g}{\pi \times R^2}. \tag{3.8}$$

Equation 3.8 implies that an increase in robot mass or a reduction of its contact surface can be encountered by an increase of differential pressure [81].

3.2.2.3 Air Velocity

I can calculate the air velocity of the outstreaming gas through the Bernoulli equation 3.9.

$$P_R + \frac{\rho_{air}}{2} \times V_R^2 + \rho_{air} \times g \times h_R = P_a + \frac{\rho_{air}}{2} \times V_a^2 + \rho_{air} \times g \times h_a = const. \quad (3.9)$$

The three terms with subscript R refer to pressure energy, kinetic energy, and potential energy *inside* and the three terms with subscript a to *outside* the pressure chamber. I assume that V_R, h_R and h_a are infinitely small. Therefore:

$$V_a = \sqrt{\frac{(P_R - P_a) \times 2}{\rho_{air}}} \quad (3.10)$$

3.2.2.4 Output Airflow

I can calculate the output airflow. Q_a is proportional to N_{holes} and r_{hole}^2.

$$Q_a = V_a \times \sum_{n=1}^{N_{holes}} A_{hole} \quad (3.11)$$

ΔP and Q_a specify the blower requirements.

3.2.2.5 Example Calculation

HoverBot version 1 weights $M_{robot} = 0.02kg$ and its radius is $R_{robot} = 0.02m$; its payload is:

$$P_{robot} = \frac{0.02\,\mathrm{kg} \times 9.81 \frac{m}{s^2}}{\pi \times (0.02m)^2} = 156\,\mathrm{Pa}$$

Therefore, the *minimum* differential pressure that is required to lift the robot from the arena surface is $\Delta P = 156\,\mathrm{Pa}$. According to Equation 3.10, the air velocity of the outstreaming gas is

$$V_a = \sqrt{\frac{(156\,\mathrm{Pa}) \times 2}{1.225\,\frac{\mathrm{kg}}{\mathrm{m}^3}}} = 16\,\frac{\mathrm{m}}{\mathrm{s}}$$

The magnet-levitation table has $N_{airholes} = 645$ with an airhole size of $d_{airhole} = 1.6\,\mathrm{mm}$. This equivalents to an outlet of size:

$$A_{holes} = \sum_{n=1}^{645} \pi \times (0.0008m)^2 = 12.97\,\mathrm{cm}^2$$

and a total output airflow of

$$Q_a = 16\,\frac{\mathrm{m}}{\mathrm{s}} \times 12.97\,\mathrm{cm}^2 = 1.24\,\frac{\mathrm{m}^3}{\mathrm{min}}.$$

The differential pressure and output flow are the key variables for choosing an air blower. In this case, the air blower has to deliver $159\,\mathrm{Pa}$ at an airflow of $1.24\,\frac{\mathrm{m}^3}{\mathrm{min}}$. These calculations are good estimates and significantly help engineering the system, however, they are approximations. While the height of the gap between HoverBot and magnet-levitation table can be determined, I developed the levitation table based on the equations of this section. More complex calculations might become important for the development of the next generation of HoverBots. Therefore, I refer the interested reader to Leal et al.'s work on thin-lubrication problems [81].

3.2.3 Magnetic Anchors

The magnet-levitation table has an array of permanent magnets embedded into its top surface. The magnetic grid serves a double purpose: i) it supplies robots with magnetic fields that they use as magnetic anchors for locomotion, and ii) it supplies robots with an energy field that they detect with their magnetic field sensors and subsequently analyse to derive information about their surroundings. I discuss the magnetic fields and their influence on HoverBot's sensing capabilities in Chapter 4. Figure 3.11 shows the top surface of the magnet-levitation table from patent application GB1611448.0 filed by the University Court of the University of Edinburgh.

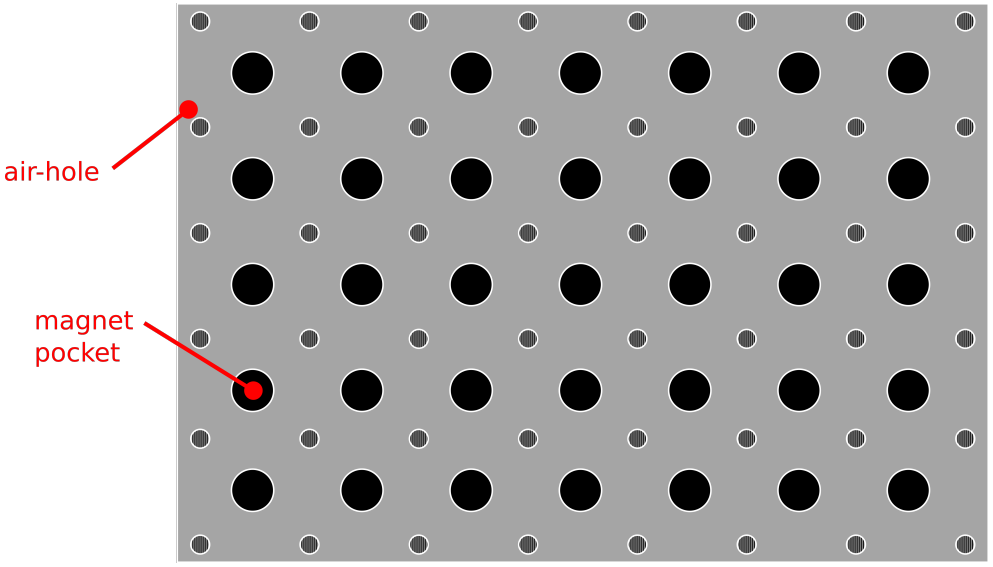


Figure 3.11: Illustration of the magnet and air-hole patterns used in the patent application GB1611448.0.

3.2.3.1 Design Criteria

Permanent magnets can vary in size and material; they can be embedded mono-directional (all north-pole facing up or all south-pole facing up) or with varying polarity (north-south-north- etc.), and they can be arranged in different patterns ranging from quadratic to radial configurations. Some of the possible implementations are more difficult to manufacture than others.

- I only embed magnets mono-directional, north-pole facing up, to reduce the manufacturing effort of the magnet-levitation table.
- I choose a quadratic magnetic grid with N^2 magnets to simplify HoverBot's electro-magnetic interactions with the permanent magnets.

While other configurations could be engineered, they do not align with the core motivation of my work: the design of swarm robotic technology that is easily manufacturable.

3.2.3.2 Magnetic Grid Pattern

Figure 3.12 shows the side view of a HoverBot levitating on the magnet-levitation table. Let d_{mm} be the distance between adjacent permanent magnets, d_{cc} the distance between two robot coils, and S be the normalised distance (shift) between coils and magnets.

$$S = \frac{d_{mm} - d_{cc}}{d_{cc}} = \frac{d_{mm}}{d_{cc}} - 1, \quad d_{mm} = d_{cc}, ..., 2d_{cc} \tag{3.12}$$

The smaller the distance between two coils, the more space efficient is the robot design. The minimum distance between two robot coils is the coil diameter itself $d_{cc,min} = \oslash_{coil}$.

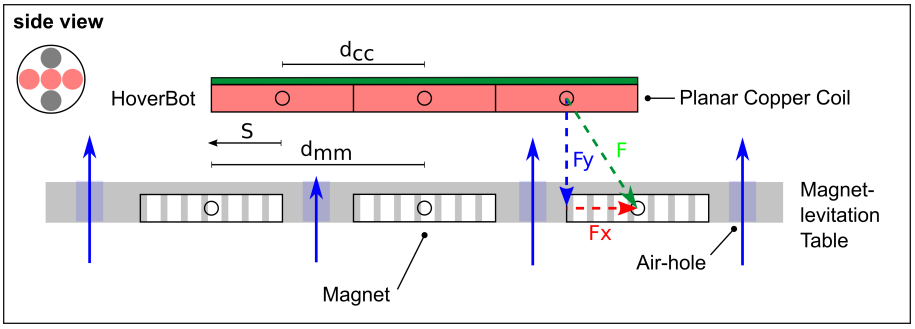


Figure 3.12: Conceptual figure: the dimensions of objects are not proportional. The electro-magnetic field evoked by a robot coil interacts with the magnetic field of a permanent magnet. The distances between coils and magnets determine the actuation pattern.

Figure 3.12 also indicates the magnetic interaction between a robot coil and permanent magnet. The magnetic field vector $\vec{F}$ is broken down into its components - F_x and F_y. $|F|$ qualitatively indicates the effort for a robot to pull itself towards an adjacent magnet. $|F|$ increases with greater S. S defines the spatial relationship between coils and magnets. For each S, the coil-magnet interaction becomes different.

No Movement $d_{mm} = d_{cc} \rightarrow S = 0$

Coils and magnets are aligned as illustrated in Figure 3.13a. In this configuration $F = F_y$; coil actuation causes movement in y-direction, hence the robot cannot move, but reduce the gap between itself and the arena surface.

Deterministic Movement $d_{mm} = 1.5 \times d_{cc} \rightarrow S = 0.5$

Coils and magnets overlap by 50% as illustrated in Figure 3.13b. In this configuration, F is shorter to adjacent magnets than to the centre magnet - green (F_1) vs. pink (F_2) trajectories. Coil actuations are guaranteed to follow green trajectories; robot movement is deterministic.

Probabilistic Movement $d_{mm} = 2 \times d_{cc} \rightarrow S = 1$

Coils and adjacent magnets do not overlap as illustrated in Figure 3.13c. In this configuration, F is equally long to adjacent and centre magnets - green and pink trajectories are equally long. Coil actuations could follow either trajectory; this movement is probabilistic.

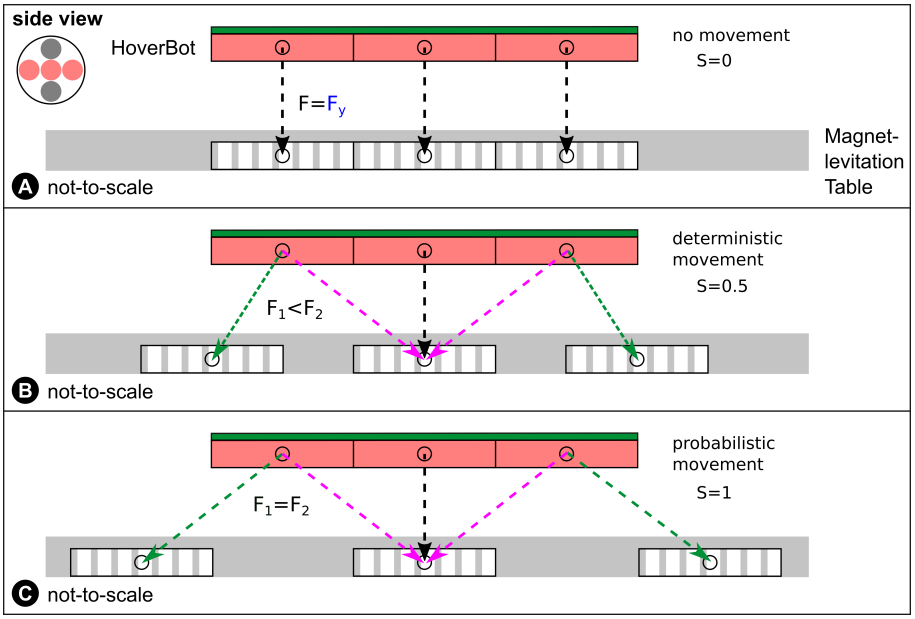


Figure 3.13: Conceptual figure: the dimensions of objects are not proportional. The magnet-coil spatial relationship is determined by shift S. A) If there is no shift between coils and magnets, coil actuation does not lead to movement. B) If there is a 50% shift between coils and magnets, coil actuation leads to movement in one of two directions at any given time. C) If there is a 100% shift between coils and magnets, coil actuation has vague outcomes; the robot could move either direction dependent on the slightly shorter F (green or pink) at the time of actuation.

3.2.3.3 Full Actuation Cycle

Figure 3.14 shows the full actuation cycle of a HoverBot and a coil-magnet configuration for $S = 0.5$. This setup establishes a 50% overlap between two coils and two magnets at any given locomotion step. HoverBot always moves towards the magnet with shorter $|F|$ which are marked as green trajectories. This constellation allows a three-step-movement; it takes HoverBot three coil actuations to move from one magnet to another.

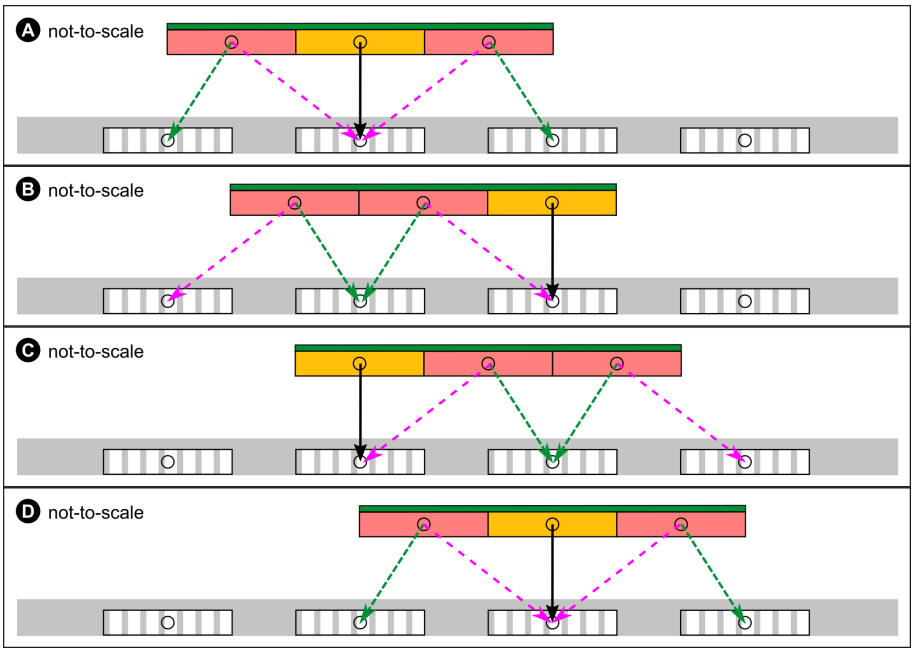


Figure 3.14: Conceptual figure: the dimensions of objects are not proportional. Cross-section of the magnet-levitation table and a hovering robot. In this setup, magnets and coils overlap 50% with one another. Actuated coils are coloured orange. A movement consists of three steps, whereas each step can be reversed at any time. The steps are equidistant. This pattern allows movements in all four cardinal directions [North, South, East, West].

The advantages of the $S = 0.5$ configuration are:

- A movement consists of three steps, whereas steps are equidistant.
- Any step can be reversed at any given time.
- If the centre coil aligns with a magnet, all four side coils overlap equally with adjacent magnets.
- $|F|$ is identical for each step.

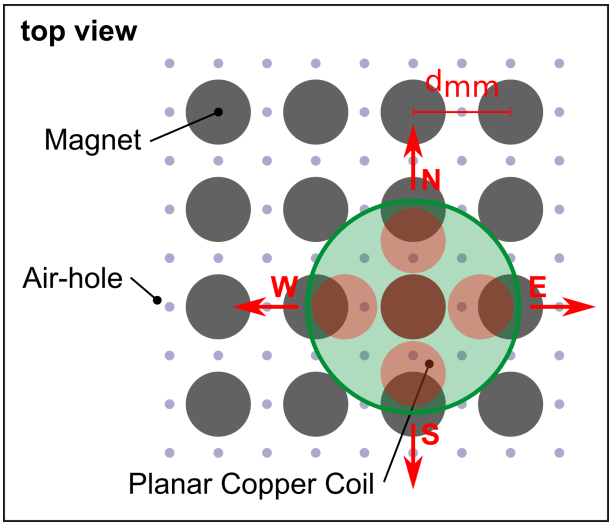


Figure 3.15: Top view of a HoverBot on a magnet-levitation table with magnet-to-coil shift $S = 0.5$. If the centre coil aligns with one of the magnets, all four side coils align equally (50%) with adjacent magnets. The step size for any coil actuation is the same.

Figure 3.14 shows the coil actuation sequence for one-dimensional movement. This actuation scheme can be extended to two-dimensional movement as depicted in Figure 3.16. This magnetic grid design allows robots to move in all four cardinal directions by actuating their coils with one-directional currents, and by embedding magnets mono-directional. It makes robot designs simpler and magnet-levitation table manufacture easier.

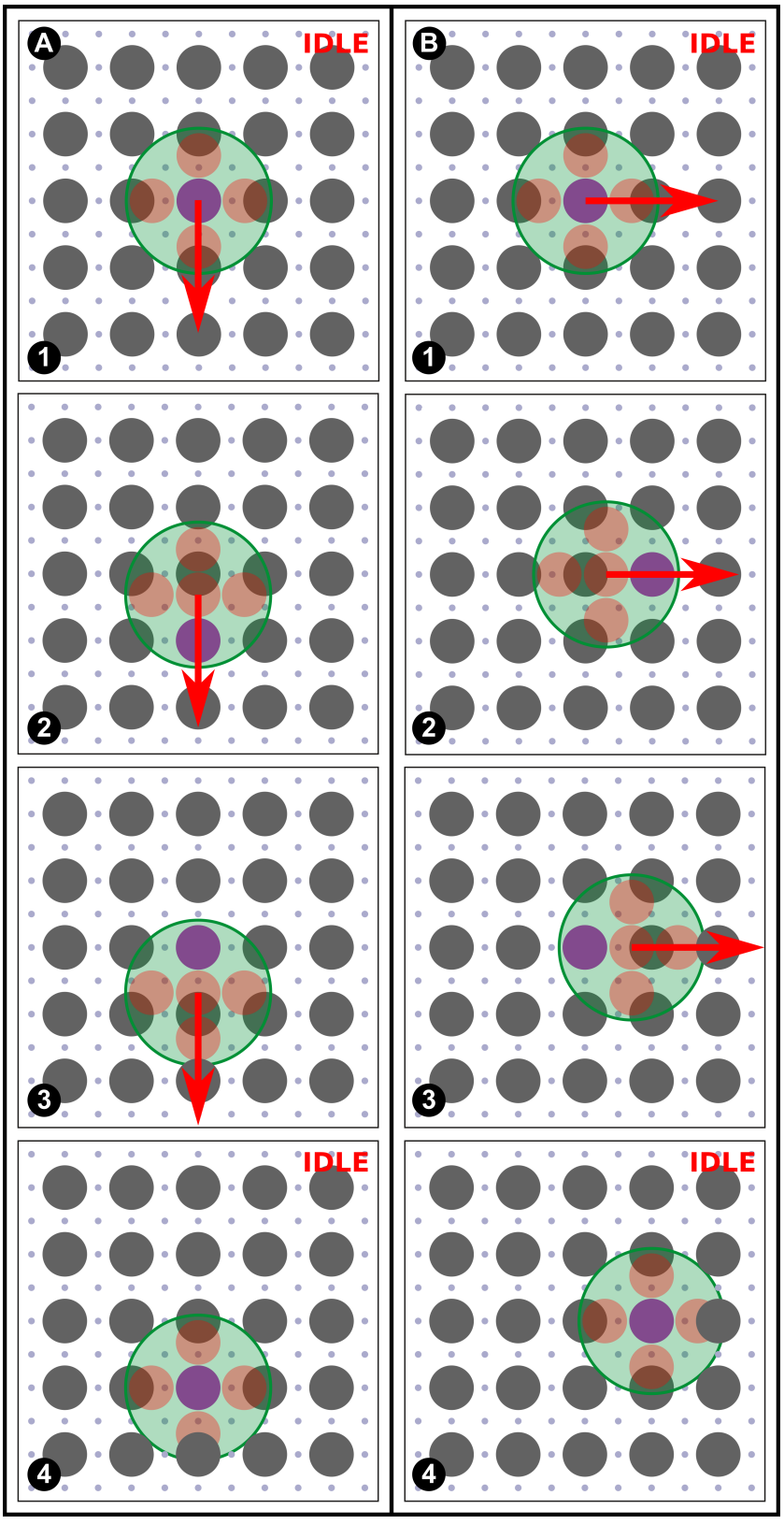


Figure 3.16: A) Movement to the South: 1) The robot starts in idle state, that is, the centre coil aligns with a permanent magnet. The robot actuates its bottom coil; 2) its top coil; and 3) its centre coil; 4) the robot is back in idle state. B) Movement to the East: 1) The robot starts in idle state. The robot actuates its right coil; 2) its left coil; and 3) its centre coil; 4) the robot is back in idle state.

3.2.3.4 Choosing Magnets

Permanent magnets possess a variety of attributes. They can vary in i) shape (circular, quadratic, rectangle, spheric, rings, etc.); ii) size (a few mm to many cm); iii) material (neodymium, alnico, samarium cobalt, ceramic, etc.) determining the magnetic strength of the magnet; iv) coating to protect the magnet from corrosion (nickel, zinc, tin, copper, epoxy, etc.); v) being axially or diametrically magnetised; vi) coercivity (determining the resistance against demagnetisation); vii) and cost ranging from few to hundreds of pounds sterling. The most important criteria for my application are: i) size; ii) shape; iii) strength; and iv) cost. If the magnets are too large, I will not be able to build a vast magnetic grid; smaller magnets allow higher degrees of integration. If the magnets are not in disc shape, the circular robot coils will not perfectly align with the magnets potentially leading to movement imperfections. If the magnets are not sufficiently strong, if they cannot penetrate the arena substrate, HoverBots will not be able to interact with them, hence, will not be able to move. If the magnets are too expensive, I cannot purchase them, hence I cannot build the magnet-levitation table.

I evaluated whether the size of disc magnets enhances HoverBot's locomotion. I purchased 12 magnets of each category that is listed in Table 3.2. I did not observe locomotion improvements with increasingly larger magnets; robot locomotion was independent of magnet size. Therefore, I purchased the lowest cost permanent magnets ($\pounds 0.08$ per magnet - Amazon UK): neodymium (N42) magnets with a diameter of $10\mathrm{mm}$ and a thickness of $3\mathrm{mm}$.

Diameter	Thickness	Material	Article Number	Cost
$9.5\mathrm{mm}$	$6.3\mathrm{mm}$	N42	D64	£0.79
$11.1\mathrm{mm}$	$6.3\mathrm{mm}$	N42	D74	£1.00
$12.7\mathrm{mm}$	$6.3\mathrm{mm}$	N42	D84	£1.25

Table 3.2: List of permanent magnets that I purchased from KJ magnetics to quantitatively evaluate whether magnet size impacts robot locomotion. US Dollar to British Pound currency rate: 1.32 (07/2018).

3.2.4 Implementation

While I qualitatively describe the fluid mechanics and the magnetic grid of the magnet-levitation table in Sections 3.2.2 and 3.2.3, this section deals with the quantitative implementation of the magnet-levitation table. In Section 3.2.4.1, I give an overview of the magnet-levitation table design. In Section 3.2.4.2, I explain the design choices that create sufficient uplift to make HoverBots levitate. I describe the best practise of how to embed magnets into the top-surface of the magnet-levitation table in Section 3.2.4.3. In Section 3.2.4.4, I talk about the inclination of the magnet-levitation table; HoverBots could experience movement issues if the magnet-levitation table is not levelled.

3.2.4.1 Design Overview

Figure 3.17 shows the most-recent design of the magnet-levitation table. The table is made of MDF board and consists of three different shapes: one bottom layer, one top surface, and four side-walls. The edges of each layer have finger joints to increase the stability of the table. The bottom layer is blank; the top-layer has an array of air-holes and magnet pockets milled into its top surface; and each side wall has a fan inlet.

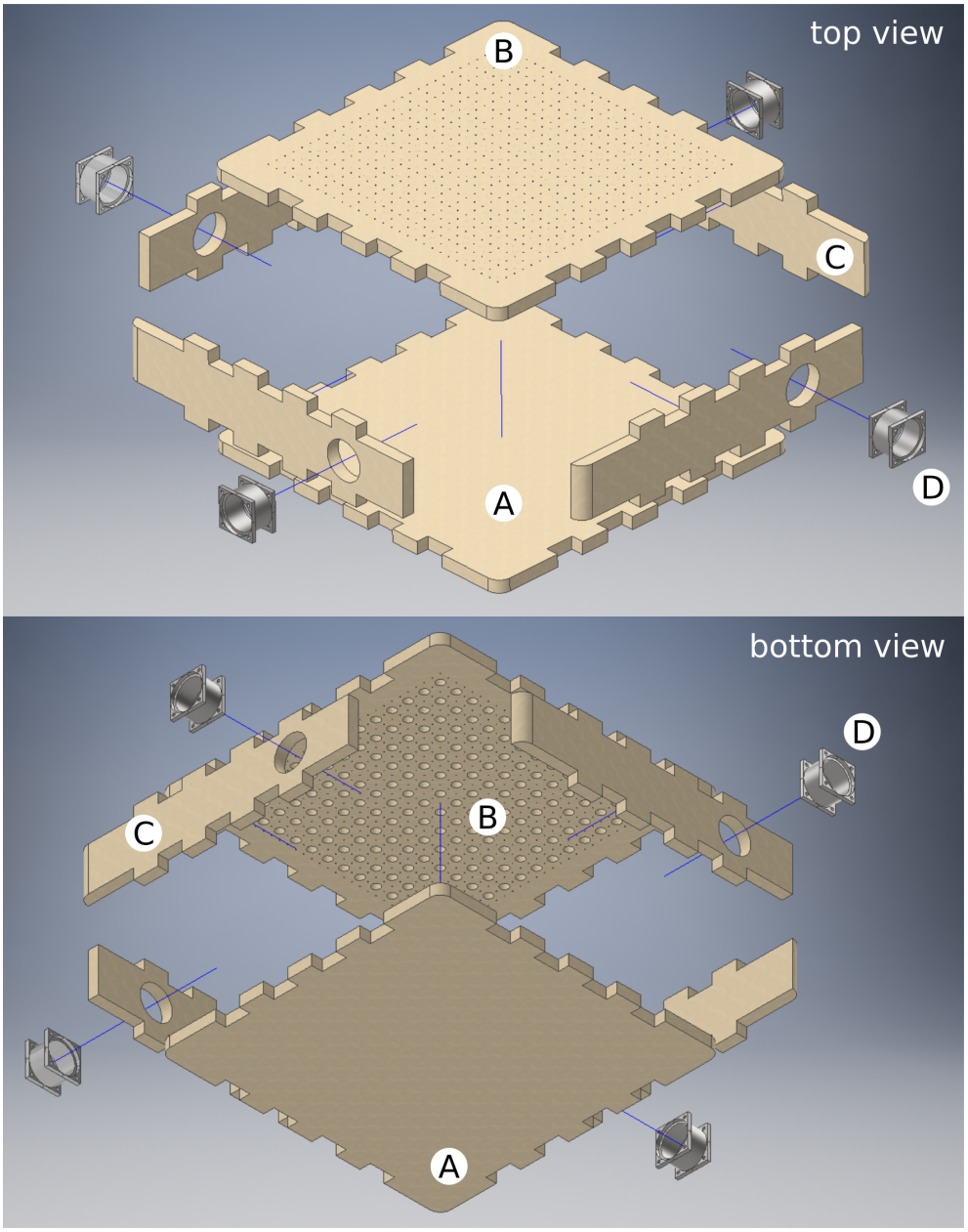


Figure 3.17: The magnet levitation table consists of: A) bottom layer; B) top surface; C) four side walls; and D) four fans. The side walls are inserted into the bottom layer and the top surface closes the chamber. Appendix C contains the technical drawings of the magnet-levitation table.

3.2.4.2 Fluidic System

The air-blower of the magnet-levitation table must supply a differential pressure of 159Pa and an airflow of $1.24\frac{\text{m}^3}{\text{min}}$ as I discuss in Section 3.2.2.5. There are many different air-blowers available, however, only a few supply high static pressure at moderate air-flow rates. Figure 3.18 shows the pressure-airflow curve of the San Ace 40 9HV0412P3K001 from Sanyo Denki. It is a 40mm direct current cooling fan for servers and communications equipment with high static air pressure at low to medium airflow rate featuring industry's highest static pressure fan at this size [82]. Figure 3.18 indicates that the static pressure airflow curve does not suffice the requirements of the HoverBot system. Even at 100% Pulse Width Modulation (12V), the air blower only delivers $0.75\ \frac{\text{m}^3}{\text{min}}$ at $175\,\text{Pa}$. Besides that, it is also generally advisable to operate fans far below their maximum specifications mainly due to noise. I started prototyping the magnet-levitation table with a conventional leaf-blower, however, the leaf blower was too noisy for a laboratory setup.

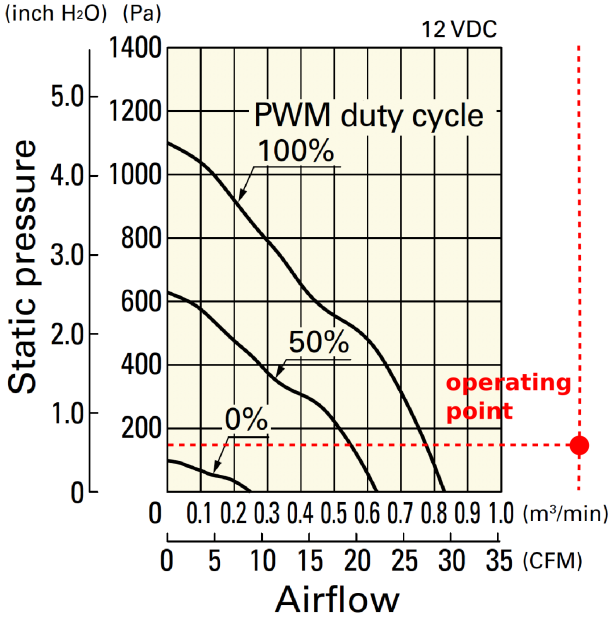


Figure 3.18: Static pressure-airflow characteristics of the San Ace 9HV04 fan. The operating point for the HoverBot system is beyond the capability of a single fan. Several fans have to be stacked in parallel to increase the input airflow rate.

Stacking Fans in Parallel and Series

While a single San Ace 9HV04 is not able to provide the airflow and static pressure to make HoverBots levitate, air sources can be arranged in series and parallel to increase static pressure and airflow. Similar to batteries, you can stack two fans in parallel to increase the air-flow (cp. current) and you can stack two fans in series to increase the static pressure (cp. voltage) of the system as illustrated in Figure 3.19 [83]. I used this stacking method to engineer a system that meets the fluid mechanic requirements. Moreover, by stacking more fans than I actually require, the HoverBot system becomes compatible for even potentially smaller or heavier HoverBot iterations as I discuss in Section 3.2.2.2. I power the fans with a Dr. Meter Triple Linear DC Power Supply 30V 5A which I purchased from Amazon USA.

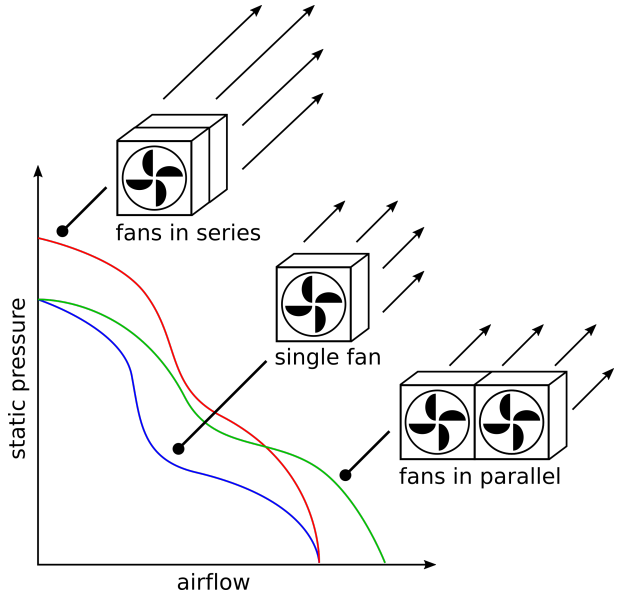


Figure 3.19: I can stack fans in series to increase the maximum static pressure or in parallel to increase the maximum airflow.

3.2.4.3 Magnetic Grid

The magnet-levitation table underwent several iterations. I identified two methods of how to embed magnets into the top layer of the magnet-levitation table. I can either drill holes or mill pockets into the top-layer. In the following, I will discuss both methods and indicate their advantages and disadvantages.

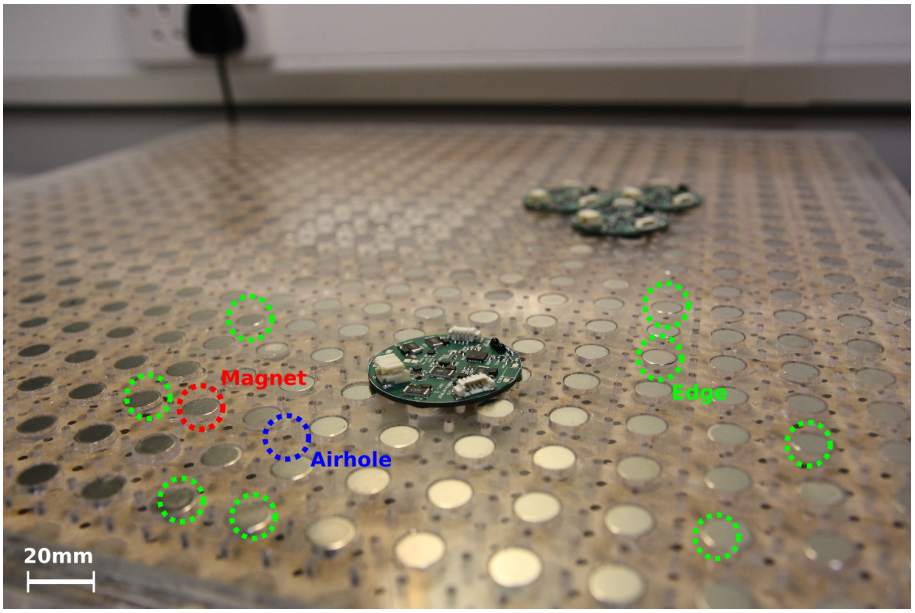


Figure 3.20: If magnets are not perfectly embedded into the top surface of the magnet-levitation table, they become obstacles to HoverBots. The magnets that are circled in green are examples of magnets that edge out. An increase in airflow could help HoverBots to overcome such obstacles, however, it is also more difficult to control HoverBots that move on a more turbulent air bed.

Magnet Holes

A laser cutter can be used as a rapid prototyping tool. It is able to cut very fast through a variety of materials which shortens fabrication time. First, I designed the magnet-levitation table for manufacture with a laser cutter which is shown in Figure 3.22a. Permanent magnet and magnet hole diameters matched (press-fit); magnets had to be forced into the magnet holes. This tight fit ensured that magnets stayed in place until they were permanently glued.

The press-fit turned out to be problematic. Magnets slightly varied in size due to manufacturing tolerances. Forcing a slightly too large magnet into a hole of non-extendible material such as acrylic causes it to crack. Substituting acrylic with MDF did not help either. Although MDF tends not to crack, the wood fibres deform under pressure and the top surface starts distorting. Any imperfection on the top surface could become obstacles to HoverBots since they only levitate on a very thin air cushion. Moreover, drilling holes through a surface implies damaging both substrate sides. Magnets have to fill up the holes, whereas every magnet must perfectly align with the top surface to produce a flat finish. Figure 3.20 shows several edges that were caused by imperfectly placed permanent magnets.

I changed the magnet fit from press-fit to clearance-fit to avoid damaging the top surface. I placed the top surface onto a steel table; the magnetic interactions between magnets and steel table dominated the magnetic interactions between adjacent magnets. Embedding magnets was a case of sliding magnets (mono-directional) into magnet holes until they made contact with the steel table and gluing them into place.

Magnet Pockets

However, table designs with magnet holes always ended up producing edges; magnets became obstacles and prevented HoverBots from moving. HoverBots require a magnet-levitation table with an undamaged and smooth top surface. Therefore, I started to mill magnet *pockets* with a CNC-machine instead of drilling magnet *holes* with the laser cutter. Pockets are cavities that are milled into the surface. They differ from holes in that they do not penetrate both surfaces of a substrate. There is usually a gap to the intact surface as illustrated in Figure 3.22b. I used an end-mill to machine flat-bottomed holes into the MDF with a drilling depth tolerance of $\Delta d = 0.2\text{mm}$. I engineered a clearance fit; magnet holes were slightly bigger than magnets.

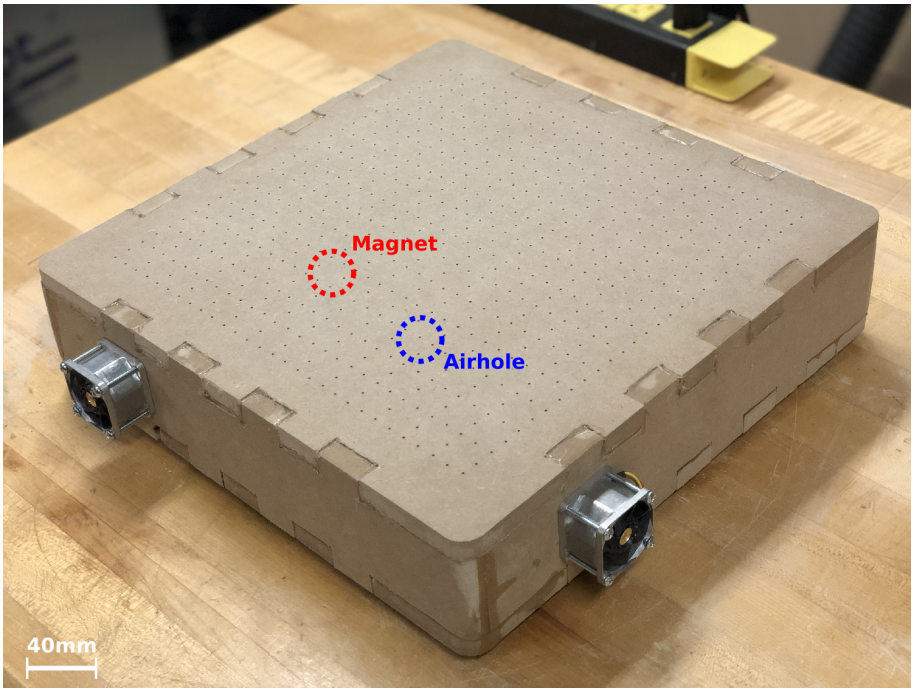


Figure 3.21: Magnet-levitation table with magnet pockets. Magnets are embedded beneath the surface. The table surface remains smooth which is important for robot locomotion.

By machining pocket holes, using clearance fit, and embedding magnets on a steel table, it is possible to manufacture a magnet-levitation table with a smooth top surface.

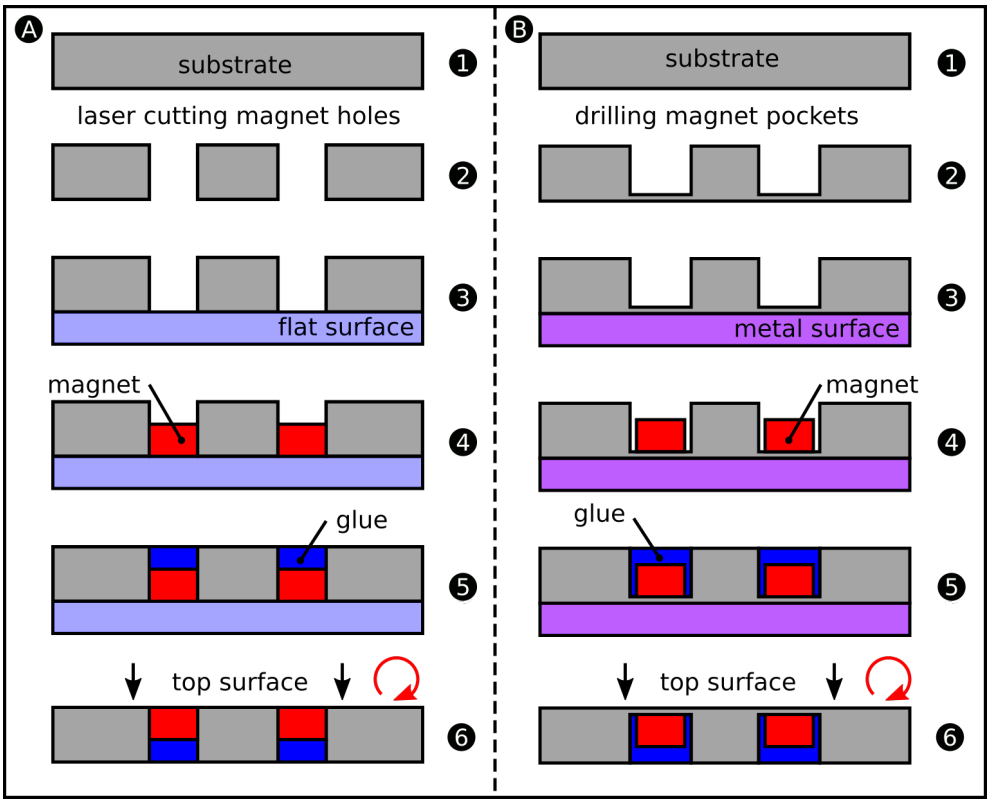


Figure 3.22: There are different techniques how to embed magnets into the top surface of the magnet-levitation table. A) 1-2) A laser cutter cuts magnet holes into the top surface; 3) the top surface is placed on a flat substrate; 4) magnets are pressed into the holes (press fit); 5) magnets are glued into place with a hot glue gun; and 6) magnets must be perfectly embedded into the top surface; in reality at least some magnets edge out and become obstacles for HoverBots. B) 1-2) A CNC-machine drills magnet pockets into the top surface, one surface side remains intact; 3) the top surface is placed on a steel substrate; 4) magnets are slid into the pockets which interact with the steel substrate (pockets are larger than magnets); 5) identical to A5. 6) Magnets are fully embedded, the top surface remains flawless.

3.2.4.4 Table Inclination

The inclination of the magnet-levitation table is paramount for operation. If the table is not levelled, HoverBots indicate difficulty moving. A mild slope could make it more difficult for HoverBots to move, a severe slope could make HoverBots slide off the platform. A three-point levelling system and a two-dimensional bubble spirit level allow to adjust the magnet-levitation table as shown in Figure 3.23. Once the magnet-levitation table is levelled, I fixed the adjustable feet with an epoxy. Before an experiment, the inclination *should* be checked; if the table was moved, the orientation *must* be checked.

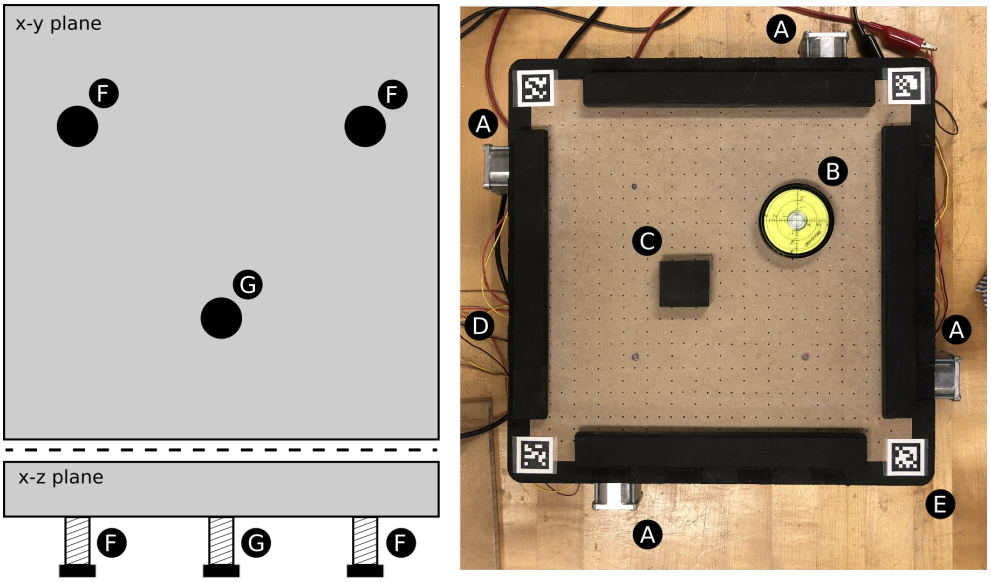


Figure 3.23: The inclination of the magnet-levitation table is paramount for operation. A slight slope makes it more difficult for HoverBots to move. A) fans; B) bubble spirit level; C) obstacle; D) wires that connect fans with an auto-transformer; E) AprilTag; and F) adjustable feet.

3.3 The HoverBot Agent

HoverBot is a printed circuit board that hovers on an air-cushion and interacts with permanent magnets; both resources are supplied by the magnet-levitation table. I discuss the magnet-levitation table in Section 3.2. Figure 3.24 illustrates the first untethered HoverBot prototype.

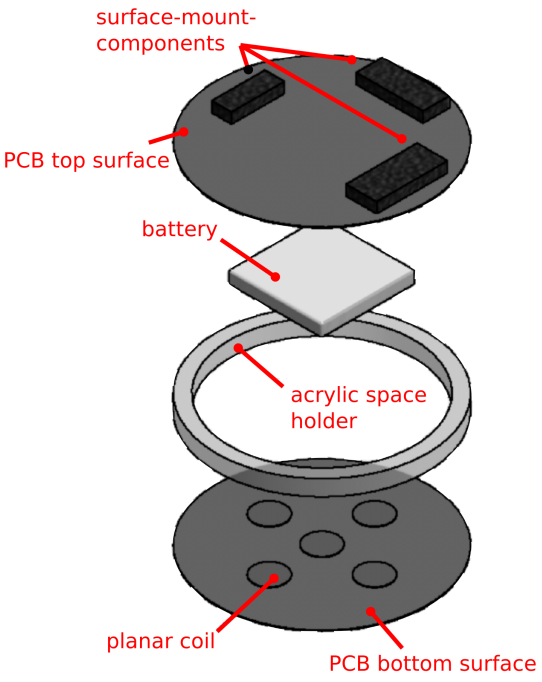


Figure 3.24: Illustration of the untethered HoverBot prototype (see Section 3.3.2.2) which is used in patent application GB1611448.0. filed by the University Court of the University of Edinburgh.

In Subsection 3.3.1, I make general comments on the design of manufacturable hardware and I explain my design principles. In Subsection 3.3.2, I give an overview of the five robot versions that I have built throughout my studies: the tethered and untethered prototypes, HoverBots versions 1 and 2, and the HoverTag. Every new robot is a successor of a previous robot design. Figure 3.25 shows the hardware block diagram of HoverBot version 2; the numbers in the figure indicate in which sequence I discuss the components. In Subsection 3.3.3, I explain my microcontroller choice and briefly discuss

HoverBot's software. In Subsection 3.3.4, I explain the actuation system. I calculate and visualise the magnetic field of a planar coil in Subsection 3.3.4.1, discuss my planar coil design in Subsection 3.3.4.2, explain the actuation circuitry in Subsection 3.3.4.3, and the coil actuation sequence to make HoverBots move in Subsection 3.3.4.4. In Subsection 3.3.5, I discuss my sensor choices. I describe the communication system and visual outputs in Subsection 3.3.6. In Subsection 3.3.7, I describe the power system design. In Subsection 3.3.8, I discuss details about the PCB design.

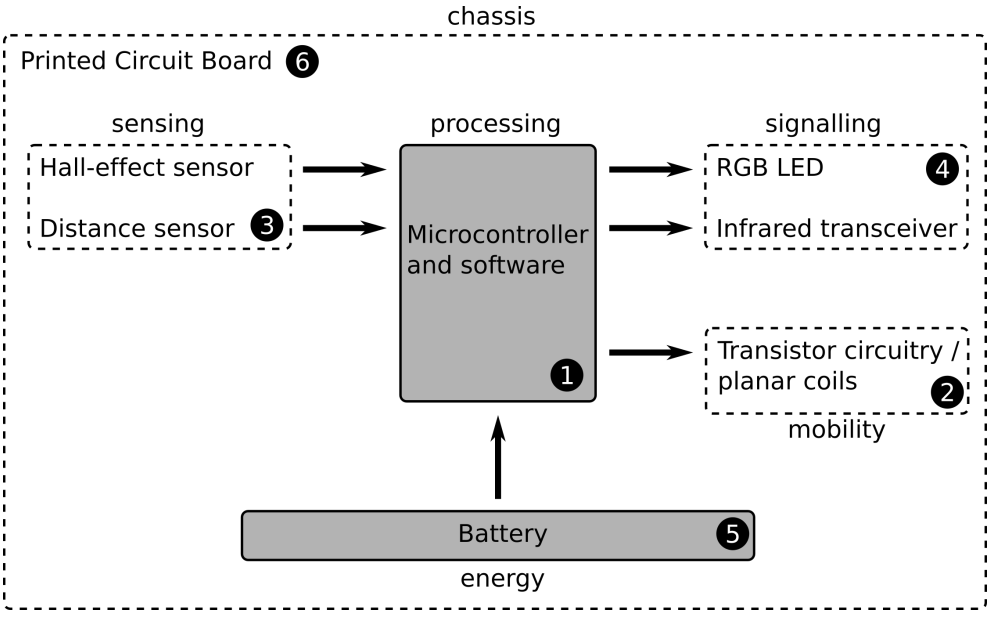


Figure 3.25: HoverBot overview. HoverBot version 2 possesses a Hall-effect sensor to detect magnetic fields, a distance sensor to detect objects above the robot, a microcontroller that processes all inputs and outputs, a RGB LED to output visual signals, an infrared transceiver for global communication, a transistor circuit to control the planar coils, and a battery. The numbers indicate the sequence in which I introduce the various components in this section.

3.3.1 Cost-Functionality Considerations

The development of swarm robots is a cost-functionality problem. The main guiding principle is scaling. Every design choice undergoes an evaluation process in which I ask myself, what impact does this design choice have on scaling up the system to hundreds or thousands of agents. This thought process changed the way I design robots.

Every electronics component can be ordered in small or in large quantities. The price breaks are usually at 10, 100, 500, and 1000 pieces. When I look for components, I always choose the lowest-cost component that also satisfies the functional requirements. It is also an useful exercise to find several different suppliers for components to avoid shortages and prevent re-designing circuitry.

Every component on a robot not only increases its cost, but also occupies space on the circuit board. It is therefore worthwhile to look for specialised components that are meant for the task at hand. For example, the actuation circuit of HoverBot version 1 consists of a MOSFET, two current limiting resistors, a smoothing capacitor, and a flyback diode. The actuation circuit of HoverBot version 2 consists of an H-bridge driver, a smoothing capacitor, and a current limiting resistor. The flyback diode is integrated into the chip and the driver does not require a gate resistor. I reduced the number of components from five to three while adding functionality; the transistor circuit only facilitates one-directional currents, the H-bridge driver facilitates bi-directional currents. I discuss both actuation circuitries in Section 3.3.4.3.

In general, every integrated circuit (IC) has a minimum circuit which is usually depicted in the datasheet. The minimum circuit often consists of additional smoothing capacitors to ensure a stable voltage supply. Capacitors filter high-frequent current peaks to prevent damage to the IC. However, if the minimum circuit of an IC requires too many additional components, I sometimes look for alternatives.

3.3.2 Implementation History

HoverBot is a populated four-layer printed circuit board with a lithium polymer battery. Its functionality comes from its surface mount components and array of planar coils. Figure 3.26 shows the various HoverBot iterations starting from the first tethered proof-of-concept prototype to the populated-by-a-PCB-foundry HoverBot version 2.

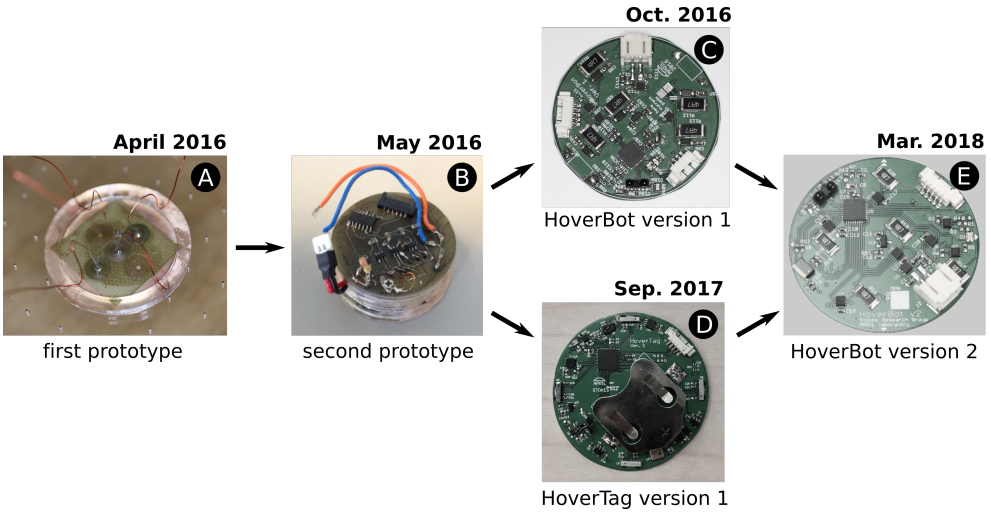


Figure 3.26: History of the HoverBot agents. A) First tethered prototype (April 2016). B) First untethered Prototype (May 2016). C) First professionally manufactured HoverBot (October 2016). D) HoverTag: a derivative of HoverBot which I developed to study the usefulness of additional sensors and to test a charging mechanism (September 2017). HoverBot version 1 was used for the development of robust actuation schemes for locomotion. E) HoverBot version 2 combines HoverBot version 1 and HoverTag. It is the most-recent HoverBot design (March 2018).

3.3.2.1 Tethered Prototype

The first HoverBot was a tethered prototype which I built as proof-of-concept system. I wanted to evaluate whether the magnetic grid pattern that I discuss in Section 3.2.3.3 was actually working. The robot consisted of a single FR4-Cu layer and an acrylic frame which is shown in Figure 3.27. I micro-machined an array of five copper coils into the copper with a LPKF Protolaser U3. I put the board design in Appendix A. The coils had a trace width and trace spacing of $50\mu\text{m}$ and 20 turns. The acrylic frame straightened the FR4-Cu layer; the copper layer curved due to its thickness. The five coils shared a common ground. I wired each coil in series to a mechanical switch. The mechanical switch applied a voltage to the coil if actuated and the coil induced a magnetic field due to current flow. The following qrcode demonstrates my first experiment with the untethered prototype which motivated all my subsequent work.

Tethered prototype proof-of-concept

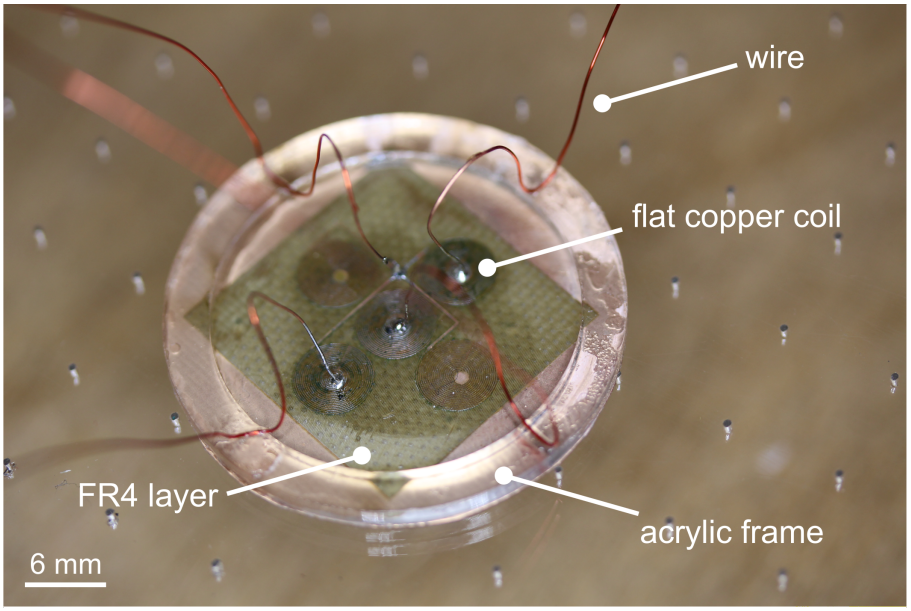


Figure 3.27: Tethered HoverBot prototype.

3.3.2.2 Untethered Prototype

I subsequently developed an untethered HoverBot prototype to proof that my technology can also operate independently/autonomously. I kept the coil layer from the tethered prototype and built another PCB that I stacked on it. The new PCB contained a ATTiny84 microcontroller, a battery connector, a voltage regulator, a programming port, a transistor circuit, and pin connectors for the coils. I put the board design and schematic in Appendix A. I manually wired the bottom PCB coils to the top PCB pin connectors. The gap between top and bottom layer is occupied by a $300\mathrm{mAh}$ lithium polymer battery. I glued an acrylic ring between top and bottom PCB to stabilise the robot chassis. Figure 3.28 shows the tethered prototype implementation. The following qrcode demonstrates my first experiment with the tethered prototype.

Untethered prototype proof-of-concept

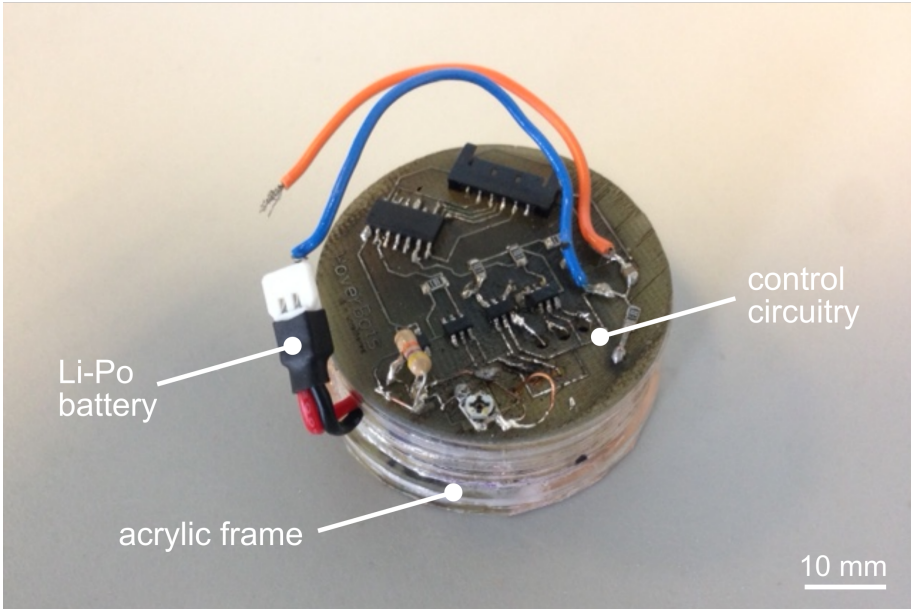


Figure 3.28: Untethered HoverBot prototype.

3.3.2.3 HoverBot Version 1

After I successfully demonstrated autonomous movement with the untethered HoverBot prototype, I designed a HoverBot version that had all its components embedded onto a single PCB, so it could be send off to a PCB foundry for populated manufacture. HoverBot version 1 consists of a single four-layer PCB, shown in Figure 3.29, and a detachable 300mAh lithium polymer battery. The bottom layer comprises five planar actuation coils. Each HoverBot has a diameter of 39mm and weighs 19.4g with, and 7.4g without, a battery. HoverBot version 1 possesses a low-power microcontroller, programming and debug ports, an infrared transceiver, a Hall-effect sensor, and a transistor circuit. The following qrcode shows HoverBot version 1 during movement.

HoverBot version 1 movement

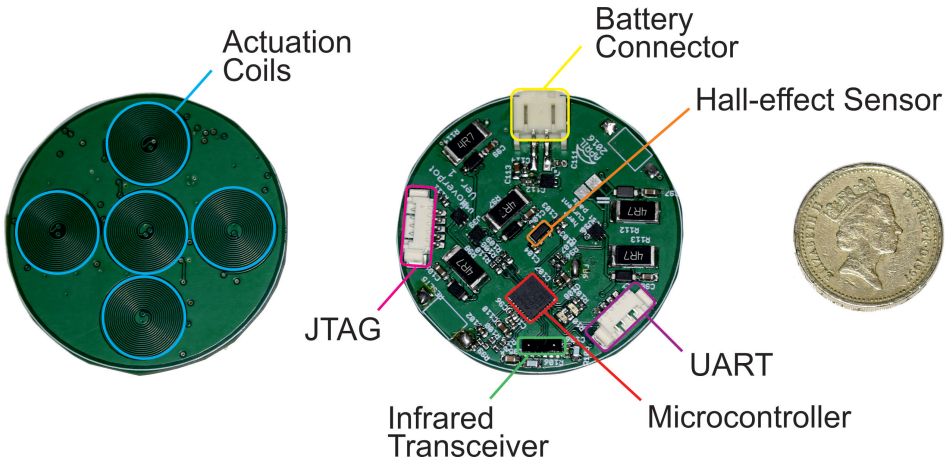


Figure 3.29: HoverBot version 1. The bottom layer of HoverBot consists of an array of five planar actuation coils. Its top layer is populated with Hall-effect and infrared sensors and a low-power microcontroller. The battery of the HoverBot is detached in this figure.

3.3.2.4 HoverTag

After demonstrating successful movement with HoverBot version 1, I considered HoverBot's locomotion mechanism to be sufficiently developed. Now I was wondering which additional sensor(s) would open up the most interesting swarm robotics experiments. I designed a new PCB that contained a diverse set of sensors and a charging mechanism and called it Hover-*Tag*. I derived this name from Texas Instrument's Sensor*Tag* which is a multi-functional sensor node. Figure 3.30 illustrates the HoverTag. I kept the infrared transceiver, microcontroller family, and Hall-effect sensor from HoverBot version 1, and added a proximity sensor pointing upwards the ceiling, a temperature sensor, an optical sensor that discriminates between red, green and blue lights, a RGB LED, a microphone, and a three-dimensional accelerometer and gyroscope. I changed the lithium polymer battery to a lithium ion coin battery. The battery holder is a surface mount component, hence, can also be populated with pick-and-place machines. I embedded a charging pad into the bottom PCB layer. A charging circuit is connected to the charging pad and battery which allows the robot to charge without removing the battery from the robot.

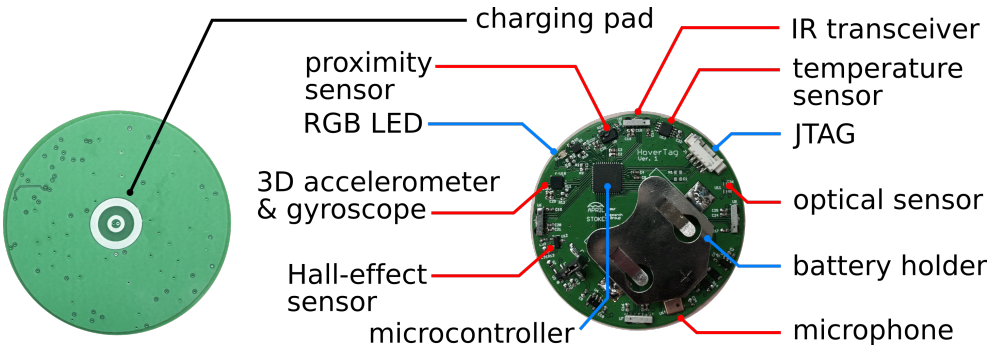


Figure 3.30: HoverTag.

3.3.2.5 HoverBot Version 2

HoverBot version 2 is the final robot iteration that I built during my studies and is shown in Figure 3.31. It is a combination of HoverBot version 1, HoverTag, and additional modifications to the actuation circuitry.

I kept the infrared transceiver, microcontroller, and battery from HoverBot version 1; the proximity sensor and RGB LED from HoverTag; and changed the actuation circuitry from MOSFET Dual N-channels to H-bridge drivers. I decided that the proximity sensor pointing upwards the ceiling would open up the most novel set of experiments (see future work in Chapter 7). The H-bridges facilitate bi-directional current flows; HoverBot's coils are now capable of inducing alternating (north and south pole) magnetic fields. The number of possible magnet-coil interactions increase. My laboratory ordered 12 populated HoverBots version 2. We used my PCB designs and mailed the electronics components to a PCB foundry. They manufactured the PCBs and populated the boards. Figure 3.32 shows an array of assembled HoverBots version 2. These are the robots that my colleagues are going to use to continue my line of research.

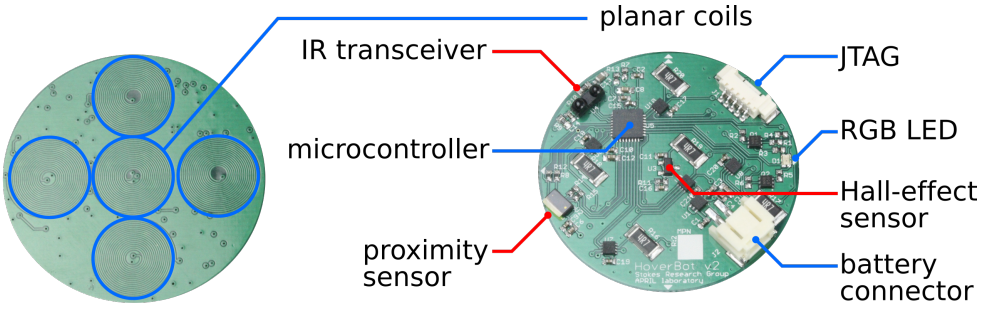


Figure 3.31: HoverBot version 2.

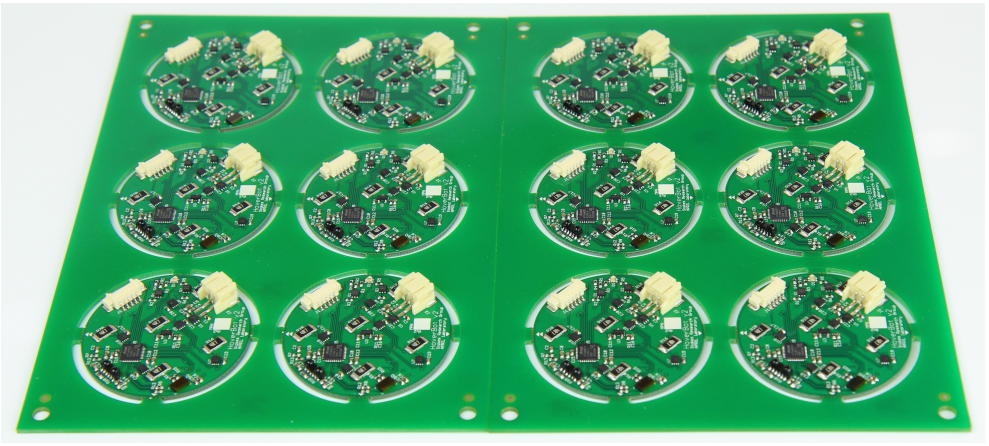


Figure 3.32: Assembled HoverBots version 2. These populated boards came directly from the PCB foundry.

3.3.3 Choice of Microcontroller and Software Design

I embedded an Atmel SAMD21E in HoverBot versions 1 and 2, and an Atmel SAMD21J in the HoverTag. The SAMD21 is a series of low-power microcontrollers using the 32-bit ARM Cortex-M0+ processor, and ranging from 32- to 64-pins with up to 256KB Flash and 32KB SRAM. The SAMD21 operate at a maximum frequency of $48\mathrm{MHz}$. SAMD21 can be ordered as quad-flat no-leads (QFN) package, a standard SMD format. Figure 3.33 shows the SAMD21E18A that I embedded into HoverBot version 2. The minimum circuit consists of five smoothing capacitors.

The main reason for choosing the SAMD series was my research collaboration with the APRIL laboratory at the University of Michigan. Professor Edwin Olson and his graduate students have implemented this microcontroller series in various projects and I had access to example code and student support during my stays in Ann Arbor. The SAMD is challenging to program mainly due to the lack of online support compared to other microcontrollers such as the AVR microcontroller family. However, SAMD have the advantage of flexible peripherals; peripheral functions can be multiplexed amongst the microcontroller pins. Microcontrollers often times possess specialised peripherals that are assigned to specific pins which makes the board design challenging.

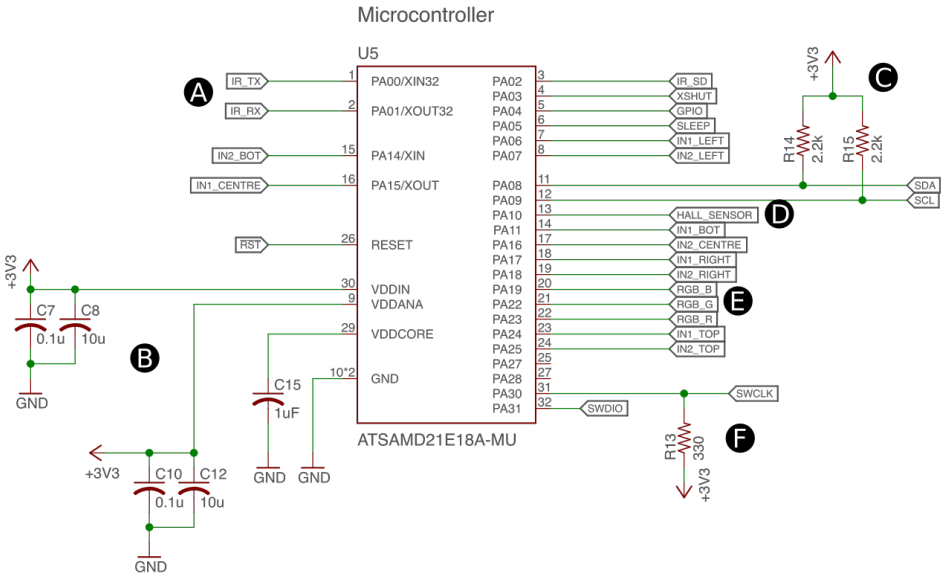


Figure 3.33: Schematic of SAMD21 microcontroller. A) infrared transceiver; B) decoupling capacitors; C) I^2C bus; D) Hall-effect sensor; E) RGB LED; and F) JTAG connector.

I programmed the SAMDs with a SEGGER J-Link JTAG/SWD Debugger. All my code is programmed in *Embedded C* and assembler language. I used a GIT repository to keep track of my code. I used the GNU software development tools which offer open source licensing and coverage of a variety of embedded processors. Access to my code is managed by my supervisor Dr. Adam A. Stokes.

3.3.4 Actuation System Design

The actuation system consists of an array of planar coils and an actuation circuit. In this section, I calculate the magnetic field of a planar coil, discuss the planar coil implementation and actuation circuitries of HoverBot versions 1 and 2.

3.3.4.1 Magnetic Field Calculations

HoverBots possess planar coils; planar coils are two-dimensional circular conductor loops. The magnetic field of HoverBots' electro-magnetic actuators can be approximated. It is dependent of i) the amount of current; ii) the number of turns; iii) and the coil geometry such as trace width or spacing between traces. In the following, I explain how to approximate the magnetic field of a planar coil [58]. This calculation confirms the measurements that I took with a magnetometer.

Single Conductor Loop

I start by considering a single conductor loop as shown in Figure 3.34. The conductor loop is $2\pi R$ long. It is divided into infinitely small line segments dl which carry a constant current I. The segments are refereed to by polar coordinates, R being the radius of the conductor loop and φ the planar angle. The total magnetic field at point $P(r, \theta, z)$ can be determined by summing up the partial magnetic fields evoked by dl along the conductor loop.

The magnetic field at observation point P can be expressed in polar coordinates as B_r and B_z components [58], whereas the total magnetic field at point P is $\sqrt{B_r^2 + B_z^2}$.

$$B_z = \frac{\mu_0 I}{2\pi\sqrt{z^2 + (R+r)^2}}\left(\frac{R^2 - z^2 - r^2}{z^2 + (r-R)^2}E_2(k) + E_1(k)\right) \tag{3.13}$$

$$B_r = \frac{\mu_0 I}{2\pi\sqrt{z^2 + (R+r)^2}}\left(\frac{R^2 + z^2 + r^2}{z^2 + (r-R)^2}E_2(k) - E_1(k)\right) \tag{3.14}$$

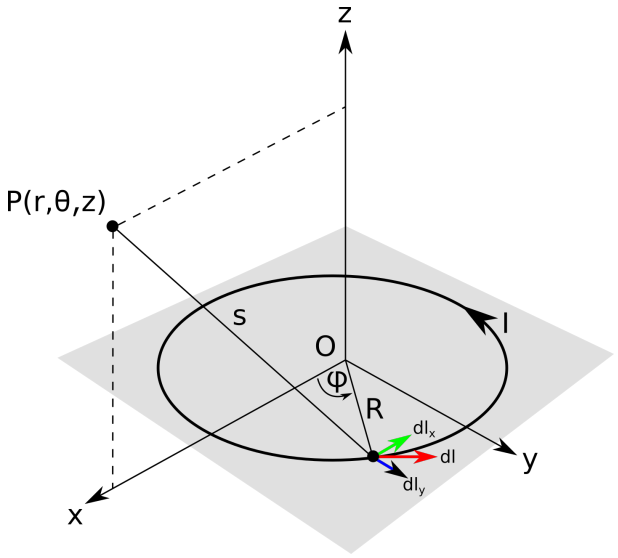


Figure 3.34: The magnetic field of a single current loop is calculated. An infinitely small line segment dl of a conductor loop carries current i. The total magnetic field at point P can be calculated by integrating the sub-magnetic-fields of dl along the conductor loop.

$E_1(k)$ and $E_2(k)$ are the complete elliptical integrals of the first and second kinds

$$E_1(k) = \int_0^{\frac{\pi}{2}} \frac{1}{\sqrt{1 - k^2 \times \sin(\varphi^2)}} d\varphi \tag{3.15}$$

$$E_2(k) = \int_0^{\frac{\pi}{2}} \sqrt{1 - k^2 \times \sin(\varphi^2)} d\varphi \tag{3.16}$$

and

$$k^2 = \frac{4 \times r \times R}{z^2 + (R + r)^2}. \tag{3.17}$$

Let μ_0 be the permeability of free space, z the height, r the radius, and φ the planar angle to point P.

Multiple Conductor Loops

I extend the system description and add several more conductor loops. The

new problem set is illustrated in Figure 3.35. Instead of having a single conductor, I have several conductors that evoke magnetic fields. The total magnetic field at point P is the superposition of all magnetic sub-fields.

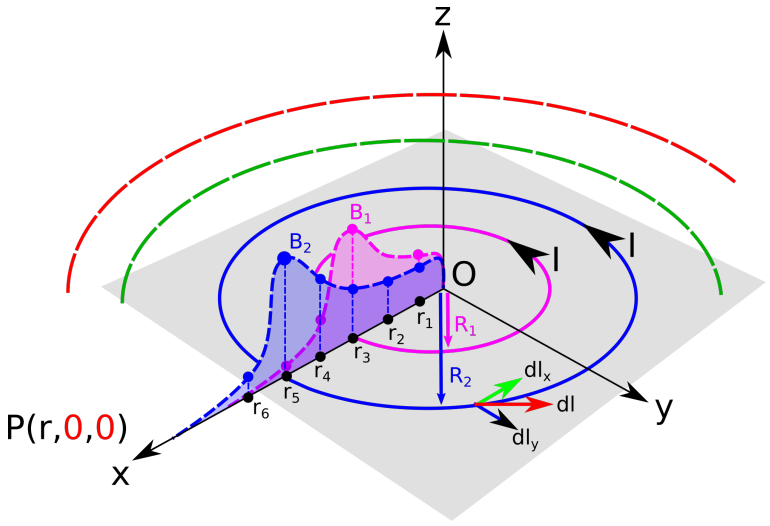


Figure 3.35: Magnetic fields induced by two current carrying copper loops with diameters $2\times R_1$ and $2\times R_2$, respectively. The total magnetic field at point P is the superposition of all magnetic fields of all current loops. I determine the magnetic field for one slice and subsequently project the slice into three dimensions by applying Cartesian-polar coordinate transformations.

The specific implementation of the theory requires assigning values to all variables of Equations 3.13 to 3.17. Table 3.3 summarises the specifications of HoverBot's planar coils. I discuss the specifications in Section 3.3.4.2. The specifications are mostly dependent of the PCB foundry's manufacturing specifications.

The coil trace are equidistant and defined as

$$R_n = R + (n-1) * (w + sp) \quad n = [1, ..., N]. \tag{3.18}$$

,whereas w is the width of a coil trace and sp the spacing between traces.

I set $\varphi = 0$, $z = 1\text{mm}$, and vary the observation point's radius from 0 to $2\times R_N$. I calculate the total magnetic fields along a straight line of observations points and sum them up to obtain the total magnetic field curve of a single slice as illustrated in Figure 3.35. I implemented this calculation in Matlab.

Variable	Value
R	2.5mm
N	17
w	$152\mu m$
sp	$152\mu m$
r	$[0, ..., 2 * R_N]$
φ	$0rad.$
z	1mm
I	500mA
μ_o	$1.2566370614 \times 10^{-6} \frac{mkg}{s^2 A^2}$

Table 3.3: Coil specifications

The symmetry of this problem allows me to project my calculation of a single slice to the third dimension. I transform the Cartesian coordinates into polar coordinates, copy the solution in increments of $\theta = \frac{\pi}{32}$ from $\theta = [0, ..., 2\pi]$, and transform them back to Cartesian coordinates. Figure 3.36 illustrates the total magnetic field shape of a planar coil with the specifications from Table 3.3. HoverBot's coil approximately induces a magnetic field of 1.4mT at a current of $I = 500\text{mA}$. I verified this calculation with a measurement using a MPU-9250 magnetometer. I measured a magnetic field strength of approximately 1.1mT in the coil's centre at $z = 1\text{mm}$.

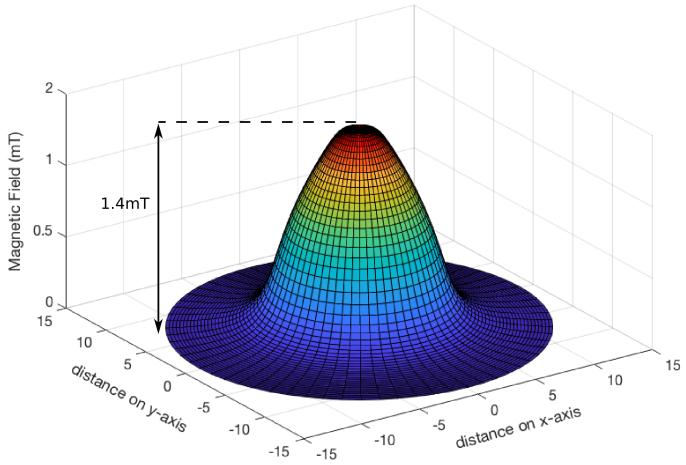


Figure 3.36: 3D magnetic field visualisation. The magnetic field decreases over distance, the magnetic field amplitude measures 1.4mT.

3.3.4.2 Planar Coil Design

I embedded planar coils in all HoverBot versions. HoverBot versions 1 and 2 are 4-layer PCBs and their planar coils are embedded into the bottom layer. In Section 3.3.4.1, I discuss the variables that impact the magnetic field of planar coils. In general, the more coil windings a coil possesses, the larger is the induced magnetic field if it is energised. I maximised the number of coil traces per unit space which is ultimately determined by the manufacturing specifications. I fabricated $50\mu\mathrm{m}$ coil traces in the HoverBot prototypes and $127\mu\mathrm{m}$ coil traces in the HoverBot versions 1 and 2, utilising the minimum specifications of the LPKF Protolaser U3 and the PCB foundry, respectively.

While choosing the minimum trace width helps increasing the number of windings, thin traces only tolerate low currents, and therewith impact the magnetic field. PCBs are FR4-Cu sheets, whereas the copper layer comes at different thicknesses. The thickness of copper in PCBs is usually measured in $\frac{oz}{ft^2}$. A $1\frac{oz}{ft^2}$ thick copper layer approximately produces $35\mu m$ thick traces. Advanced Circuits has an online trace width calculator that outputs the required trace width given a copper thickness and current [84]. The calculations are based on IPC-2221, the generic standard on printed board design.

A trace width of $150\,\mu\mathrm{m}$ and $1\,\frac{\mathrm{oz}}{\mathrm{ft}^2}$ trace thickness allow maximum currents of approximately $300\,\mathrm{mA}$ based on the Generic Standard on the Printed Board Design (IPC-2221) charts. I set the maximum current per coil to $500\,\mathrm{mA}$, which induces a magnetic field of $1.1\,\mathrm{mT}$. I measured the magnetic field by using an InvenSense MPU-9250 magnetometer. I placed the magnetometer onto the core of the centre coil. My design exceeds the suggested maximum current; I wanted to evaluate the circuitry beyond its limits. As a consequence, the planar coils dissipate heat during locomotion, the magnet-levitation table blows air beneath the robots which cools them down. In general, HoverBots should be operated below maximum current.

3.3.4.3 Actuation Circuit Design

HoverBots energise their planar coils through transistors. A transistor is an electronic switch. A microcontroller can switch transistors very quickly. The term duty cycle refers to how long a signal is active in a specified time period T. If a control signal has a duty cycle of 50%, it means that the signal is half of the time switched on and the other half switched off. The period determines whether the signal is half a second, half an hour, or half a year switched on and off. For example, If I switch a light emitting diode (LED) with a duty cycle of 50%, the period determines whether I see a blinking light once every second $T = 2\text{s}$ or whether the LED is just half as bright as it could be $T = 20\mu\text{s}$. The LED itself has a response time. If the period is too short, the component cannot react to a change. Therefore, duty cycles with short time periods are often used to control voltage amplitudes. HoverBot's microcontroller controls the power supply to its planar coils with transistors. It switches the transistors of the coils on and off. A 50% duty cycle means that the coils are powered with only half the supply voltage. This technique is refereed to as pulse-width-modulation (PWM).

In the following two sections, I discuss a couple of actuation circuitries that I have implemented in HoverBots. Please note, inductors, actuation circuitries, and magnetic fields can be studied in great depth. However, this is not the focus of my thesis.

MOSFET Dual N-channel

Figure 3.37 shows the transistor circuit that I embedded into HoverBot version 1. The circuit consists of a Toshiba SSM6N58NU Dual N-channel MOSFET which I mainly chose because of its low-cost. The transistor pair costs $\$0.59$ in small quantities (1s) and $\$0.162$ in large quantities (1000s) quantities. I put a flyback diode in parallel to the coil to prevent circuit damage from inductor discharge. Each coil is connected in series with a current limiting resistor.

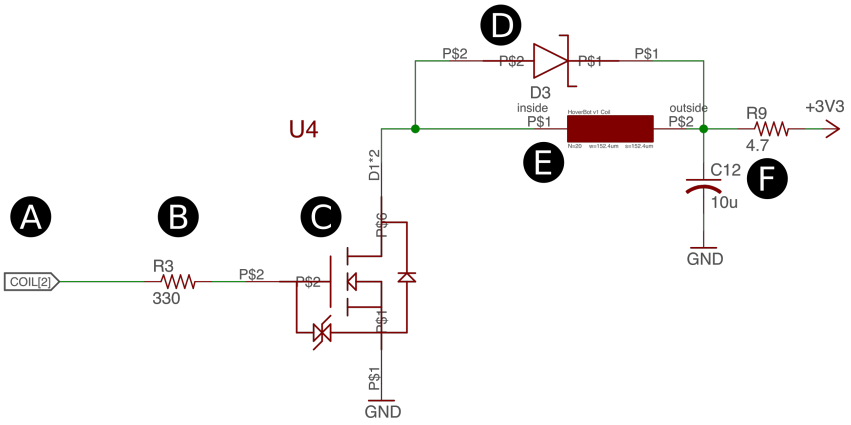


Figure 3.37: This actuation circuit consists of A) microcontroller output; B) gate resistor; C) dual N-channel MOSFET; D) flyback diode; E) planar coil; and F) a current limiting resistor and smoothing capacitor. I implemented this actuation circuitry in HoverBot version 1.

The microcontroller outputs a PWM. If the transistor is switched on, a 3.3V is applied to the planar coil in series with a current limiting resistor. The resistor ensures that the circuit cannot be driven beyond its maximum current limit. The current limit is dependent of the trace width and thickness. I discuss the PCB specifications in Section 3.3.8.

Figure 3.37a shows the microcontroller output that controls the transistor. A function outputs a PWM control signal. The microcontroller possesses five outputs to control all planar coils simultaneously. I calculate the magnetic field of HoverBot's planar coils in Section 3.3.4.1. The correct sequence of coil actuations leads to movement; I explain the actuation sequence in Section 3.3.4.4.

Robots often require calibration. For instance, a distance sensor might need calibration to assign a measurement to a distance, or vibration motors have to

be driven with motor specific currents, whereas the current might fluctuate between motors due to manufacturing tolerances. HoverBots use their planar coils to move. To ensure that all HoverBots can be driven with the same code, I measured the average series resistance of 15 actuation circuits of a total of 3 HoverBots with a Fluke 115 multimeter. The resistance of the actuation circuit can fluctuate between HoverBots. The standard deviation of the resistance was 0.1Ω which causes a current change of $7\mathrm{mA}$. Therefore, the potential current fluctuations are less than 1.5% of the total actuation current and can be neglected. HoverBots do not require actuator calibration.

H-bridge Drivers

I replaced the transistor circuit with H-bridge drivers.

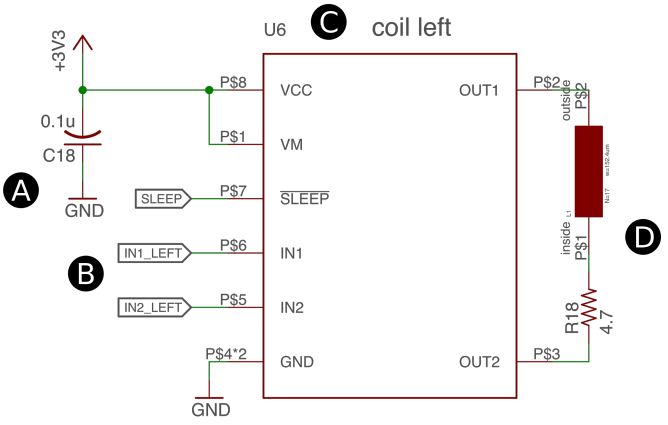


Figure 3.38: This actuation circuit consist of: A) smoothing capacitor; B) three microcontroller outputs; C) H-bridge driver; and D) a current limiting resistor in series with the planar coil. I implemented this actuation circuitry in HoverBot version 2.

I used Texas Instruments TPS73733DRVT H-bridge drivers. The H-bridge drivers have integrated flyback diodes and do not require gate resistors. This saves two components (one diode and one resistor) per coil if compared to the previous MOSFET Dual N-channel circuit. Moreover, H-bridges facilitate bi-directional current flows. HoverBots are able to induce bi-directional magnetic-fields which increase their interaction possibilities with permanent magnets. Bi-directional currents are useful to recover from rotation (locked

positions). Figure 3.38 shows the H-bridge driver. The minimum circuit consists of the chip and a single smoothing capacitor. I put a current limiting resistor in series to the coil to prevent overheating.

The microcontroller controls the H-bridge driver; the H-bridge driver possesses a digital input that can be used to switch the driver on or off. The current direction of the H-bridge driver output can be controlled by two input pins that are PWM-driven.

3.3.4.4 Coil Actuation Sequence

HoverBot levitates on air cushions and maneuvers by sequentially energizing its planar coils to pull itself toward magnetic anchors. Figure 3.39 indicates HoverBot's open-loop locomotion strategy. A single step, a movement from one magnetic anchor to another, is decomposed into three part steps. In step 1, HoverBot starts from its idle state in which its centre coil is aligned with a magnet, and the other four coils are each overlapping with adjacent magnets. HoverBot simultaneously actuates one side coil with maximum current and the opposite side coil with medium current. The maximum current is approximately $500\mathrm{mA}$ and the medium current is approximately $100\mathrm{mA}$. This actuation results in an overall movement to the right (East) while preventing HoverBot from rotating. Subsequent steps are conceptually the same, but each step requires a different pair of coils to be actuated. Three of these steps are required for HoverBot to move from one magnet to another. This actuation scheme only enables complete magnet-to-magnet movements. I have not investigated a change of direction during a part step. The relative positions of HoverBot coils and magnets are crucial for this actuation scheme. I explain the magnet-to-magnet and coil-to-coil distances in Section 3.2.3.2.

The current-time profiles can be studied in great depth. The profiles that I developed focus on reliable movement. I discuss in Chapter 7 a closed-loop control scheme that could help smooth HoverBot's movement and improve its movement velocity.

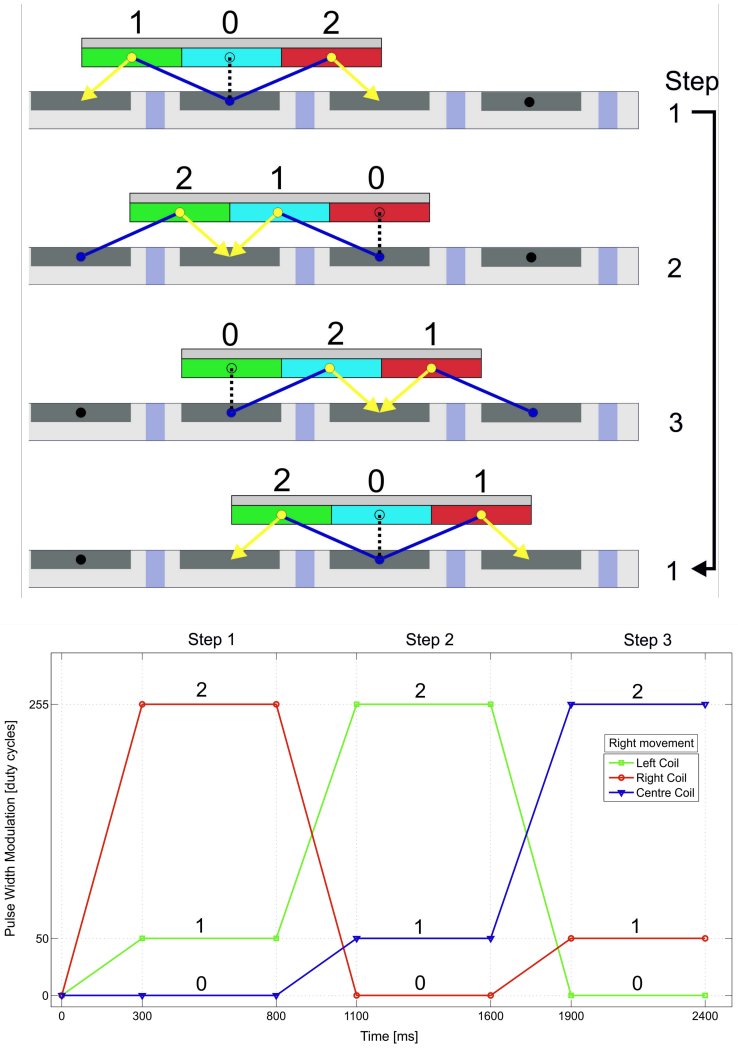


Figure 3.39: The illustrated actuation scheme describes a movement to the right (East). The other directions can be derived from this actuation profile. A HoverBot moves from one magnetic anchor to another by performing three steps. Starting in idle state (step 1), a HoverBot always actuates two coils at the time for any given step since I require two points in space to maintain the orientation of a plain. Therefore, this actuation sequence prevents rotation during movement.

3.3.5 Choice of Sensors

Sensors are transducers that in most cases transduce a physical measurement variable into an electric signal as I describe in Section 2.3.1. Analogue sensors are usually connected to analogue-digital converters. Digital sensors are either connected to an inter-integrated circuit (I^2C) or a serial peripheral interface (SPI) bus. Audio sensors depict an exception, they can be connected to an inter-IC sound bus. There are advantages and disadvantages of the various bus systems. For example, I^2C generally requires less wiring than SPI, however, I^2C is also slower in reading out sensor data. SPI requires a clock, master-in-slave-out, master-out-slave-in, and slave-select line, whereas I^2C only requires clock and data lines. I designed HoverBots as compact as possible and therefore purchased I^2C sensors. The following sensors are SMD components and summarised in Table 3.4.

HoverBot versions 1 and 2 and HoverTag possess a DRV5053OA analogue bipolar Hall-effect sensor. The sensor has a measurement range of $\pm 73\mathrm{mT}$ with a sensitivity of $-11\frac{\mathrm{mV}}{\mathrm{mT}}$. It outputs voltages from $0.2\mathrm{V}$ to $1.8\mathrm{V}$ and $1\mathrm{V}$ at $0\mathrm{mT}$. It costs $\$1.25$ in small quantities (1s) and $\$0.49$ in large quantities (1000s) if purchased from *digikey*.

HoverBot version 2 and HoverTag possess a ST VL53L0X distance sensor. It is a time-of-flight ranging and gesture sensor. It provides accurate distance measurements up to $2\mathrm{m}$. The sensor has an I^2C interface and a programmable I^2C address. It costs $\$4.50$ in small quantities and $\$2.38$ in large quantities if purchased from *mouser electronics*.

The remaining sensors are only embedded in HoverTag. HoverTag possesses a ST LSM6DS33 3D accelerometer and 3D gyroscope. The sensor has an I^2C interface. It has a full-scale acceleration range from $\pm 2\mathrm{g}$ to $\pm 16\mathrm{g}$. The sensor costs $\$3$ in small quantities and $\$1.88$ in large quantities if purchased from *digikey*. I also equipped HoverTag with a SPM1423HM4H-B digital microphone. It detects frequencies in the range from $100\mathrm{Hz}$ to $10\mathrm{kHz}$. The sensor uses pulse-density modulation to represent audio signals. The digital microphone costs $\$1.33$ in small quantities and $\$0.66$ in large quantities if purchased from *digikey*. HoverTag also has a VEML6040 RGB colour sensor with I^2C interface. The sensor is capable of discriminating between red, green, and blue lights. I installed the sensor in combination with a RGB LED. I wanted to experiment with multi-colour communication protocols. The sensor costs $\$2.52$

Sensor	Price	Technology
DRV5053OA Hall-effect sensor	$1.25($0.49)	Analogue
ST VL53L0X distance sensor	$4.50($2.38)	I2C
ST LSM6DS33 3D accelerometer	$3($1.88)	I2C
SPM1423HM4H-B microphone	$1.33($0.66)	PDM
VEML6040 RGB colour sensor	$2.52($1.06)	I2C
DS75 thermometer	$2.08($1.25)	I2C

Table 3.4: List of sensors used in HoverBot versions 1 and 2 and the HoverTag.

in small quantities and $\$1.06$ in large quantities if purchased from *digikey*. HoverTag possesses a digital thermometer. It provides digital temperature readings in a range from -55^oC to 125^oC with an accuracy of $\pm 2\%$. The sensor has an I^2C interface and costs $\$2.08$ in small quantities and $\$1.25$ in large quantities if purchased from *digikey*.

3.3.6 Communication and Visual Outputs

HoverBot version 1 and 2 and HoverTag possess infrared transceivers for communication. HoverBot versions 1 and 2 and the overhead communication system possess a Vishay TFBS4711 infrared transceiver. It is compliant with *IrDA*, a widely used wireless infrared communication protocol. It is designed for communication distances up to $1\mathrm{m}$ and supports data rates up to $115.2\frac{\mathrm{kbit}}{\mathrm{s}}$. The transceiver has a shutdown feature which cuts down its current consumption to $10\mathrm{nA}$. The microcontroller controls the infrared transceiver via specialised serial communication pins (transmit (TX) and receive (RX) lines). The infrared transceiver is a SMD component that points upwards and is used for global infrared communication. The Vishay TFBS4711 infrared transceiver has a half angle of $24\mathrm{deg}$. This infrared transceiver costs $\$4.64$ in small quantities and $\$2.9$ in large quantities. HoverTags possess ROHM RPM973-H11 infrared transceiver. They are almost identical to the Vishay TFBS4711, however, they have a 90^o mounting angle which make them ideal for local communication. The ROHM RPM973-H11 infrared transceiver has a half angle of $15\mathrm{deg}$. The infrared transceiver points sideways rather than upwards.

HoverBot version 1 possesses standard LEDs for visual signalling. LEDs are useful for debugging. HoverBot version 2 and HoverTag possess a LITE-ON LTST-C19HE1WT RGB LED. It is controlled via two Toshiba SSM6N58NULFCT

dual N-channel MOSFETs. The RGB LED can be used to output a variety of lights; each colour can be controlled individually.

3.3.7 Power System Design

The untethered protoype, HoverBots version 1 and 2 use Turnigy Nano-tech $300\mathrm{mAh}$ lithium polymer batteries. The batteries possess a discharge capacity of $20C$ which means their maximum output current is 20 times higher than their capacity ($20 \times 300\mathrm{mA} = 6A$). I charge the batteries with a Turnigy Micro-6 Lipoly Battery Charger. The batteries have to be removed from the robots to be charged. HoverBots version 1 and 2 use a Texas Instruments TPS73733DRVT voltage regulator to provide $3.3\mathrm{V}$ to the circuitry. I calculated the minimum battery life by accumulating the currents that occur during locomotion. The current locomotion strategy requires a constant current of approximately $720\mathrm{mA}$ which allows a minimum battery life of around 25 minutes. However, lithium polymer batteries should never be completely discharged due to their chemistry. A battery-watch-program monitors the battery during runtime and shuts down all circuitry when the battery reaches 90% depletion. The maximum battery life is calculated by considering HoverBot when in sleep mode, in which it approximately consumes $500\mu A$. In this low-power mode HoverBot's battery life time rises to around 600 hours or 25 days. The batteries cost $\$3.51$ a piece.

HoverTag uses a PowerStream Li-ion Coil Cell LIR2477 battery. The battery possesses a capacity of $150\mathrm{mAh}$ and a maximum discharge current of $130\mathrm{mA}$. It fits into a coin cell retainer. The battery is sufficient for testing sensors, however, the discharge capacity would be insufficient for energising planar coils. The batteries cost $\$3.44$ a piece.

3.3.8 Printed Circuit Board Design

Printed circuit boards are usually layers of fibreglass-reinforced epoxy sandwiched between thin copper layers. The fabrication technique of the PCB determines the resolution of the copper traces, the number of PCB layers, and therewith the entire circuit board design. In general, more complex circuits are easier to implement with multilayer PCBs. Multilayer PCBs are useful to keep circuit board designs compact.

The tethered and untethered HoverBot prototypes consists of a single PCB-layer. Their design is simple, and did not require a multi-layer board. HoverBot versions 1 and 2 are more complex due to increased functionality. They possess sensors, communication, and actuation circuitry which makes the PCB design more challenging.

I spent most time on aligning components during PCB design. Although there are algorithms that help routing components, manual routing remains most accurate. The microcontroller is usually the centrepiece of a PCB design; it is generally good practise to think about where to put components before start wiring them. Based on the entire list of components and the capability of the microcontroller, there might be just one configuration how to integrate all components. For example, the microcontroller I have been using in both HoverBot designs is well-known for its peripheral flexibility. Pins can be assigned to a variety of functions. However, some pins possess special functions that none of the other pins possess. In such a case, it is important to start with the most unique and continue with decreasingly rare components, whereas standard voltage output is the most basic pin function.

PCB components have a variety of attributes such as minimum voltage supply, size, or compatibility with other components amongst others. I require HoverBots to solely consist of SMD components so that they can be populated by PCB foundries. Surface Mount Devices (SMDs) are components that are directly placed onto the surface of the PCB. Non-SMD components usually require manual assembly such as through-hole components. Manual assembly defeats the purpose of HoverBots. I intend HoverBots to be purchased populated; researchers should only have to plug in a battery to the robot before starting an experiment.

3.4 Overhead Camera System

The camera system is mounted above the magnet-levitation table as shown in Figure 3.2. It is used to keep track of HoverBots and is able to draw robot trajectories such as shown in Figure 3.40. The camera system consists of a Chameleon 1.3 MP Color (Sony IXC445) camera and a Tamron 13FM28IR 2.8 mm f/1.2 day/night lens. The camera system detects AprilTags, 2D barcodes that are robust to occlusions and lens distortion [85]. Each robot possesses its own AprilTag and there are AprilTags in each corner of the table for referencing (determining HoverBots relative position). The camera is interfaced with a standard implementation of LCM, Lightweight Communications and Marshalling library [86], which provides a basic data acquisition platform for HoverBot experiments.

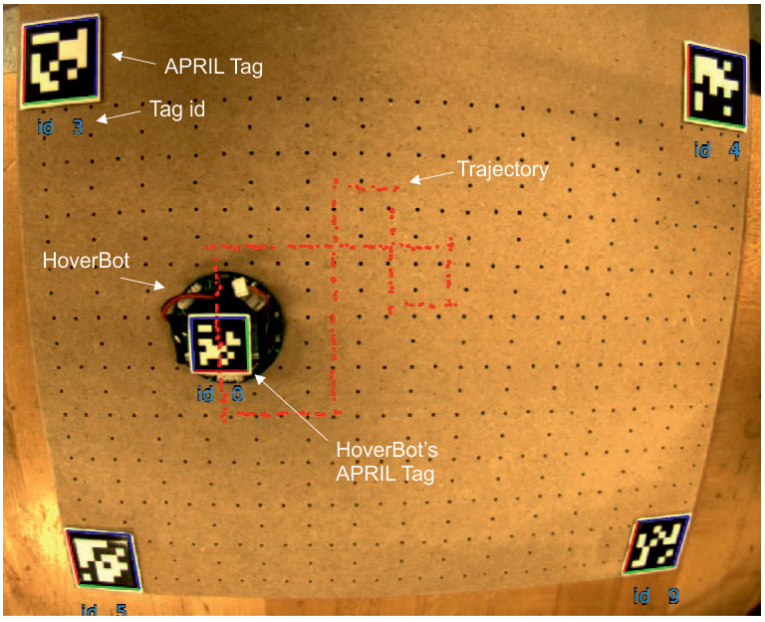


Figure 3.40: Camera setup to evaluate HoverBots during experiments. There are AprilTags placed in each corner of the magnet-levitation table and on the HoverBot itself. These tags serve as reference points and allow determining HoverBot's relative position over time. The red line indicates HoverBot's trajectory.

3.5 Overhead Infrared Communication System

The overhead infrared communication system is a custom made PCB. The communication system consists of: A) microcontroller; B) USB-to-UART chip that allows a computer to exchange messages (half-duplex) with the microcontroller; C) infrared transceiver; D) JTAG programming port; E) signalling LEDs; F) USB port to supply power and transceive messages; and G) voltage regulator. Its schematic is shown in Figure 3.41. The infrared communication system is mounted above the magnet-levitation table and has been used to transceive messages during experiments. The system is also embedded into LCM to store messages that have been passed between computer and robots. The infrared communication system is an integral part of the HoverBot system and I used it for data acquisition and debugging HoverBots.

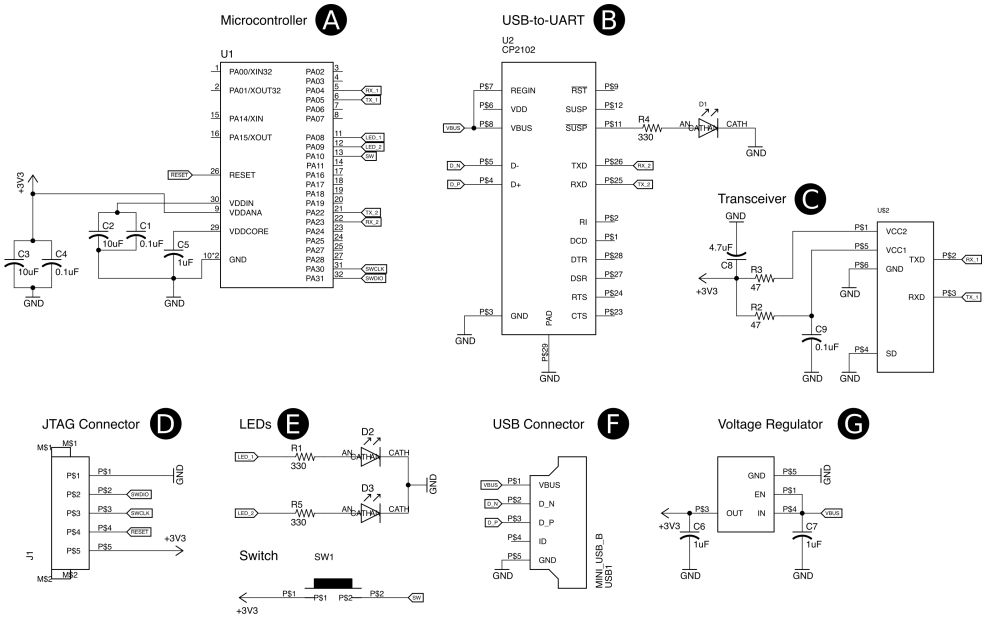


Figure 3.41: PCB schematic of the overhead infrared communication system. I have used the system to transceive messages between computer and HoverBots during experiments and to debug HoverBots.

3.6 Demonstration

In this section, I demonstrate robust movement, recovery from rotation, formation, random movement, and collisions with HoverBots version 1.

3.6.1 Movement

I tested HoverBot version 1 by conducting eleven experiments that lasted a total of 3 hours and more than 10,000 steps. In these experiments, HoverBot circled on an arbitrary trajectory until it was nearly discharged. I used a set of AprilTags to track HoverBot over time and subsequently evaluated its distance traveled, velocity, and number of missteps (errors). HoverBot performed 10,000 steps and moved with an average speed of $\mu = 0.64\frac{\text{cm}}{\text{s}}$ and a standard deviation of $\theta = 0.015\frac{\text{cm}}{\text{s}}$. I did not observe any missteps or accidental rotations during these three hours. I define a *misstep* as an unsuccessful series of energized coils that results in the robot staying on its previous position. I define an *accidental rotation* as an inadvertent robot rotation by 45^o due to local table imperfections (e.g. air flow fluctuations) or collisions with other robots or static objects. The following qrcode provides a video recording of this experiment.

HoverBot locomotion experiment

3.6.2 Recovery from Rotation

Although I have not experienced any accidental rotation incidents during three hours of testing, I developed an actuation strategy that allows HoverBot to recover from a locked position. As I discuss in Section 4.2, the Hall-effect sensor measures a local magnetic minimum if a HoverBot is locked due to accidental rotation. When a HoverBot recognizes this state, it can execute a recovery actuation scheme. The following qrcode shows HoverBot recovering from rotation.

HoverBot recovering from rotation

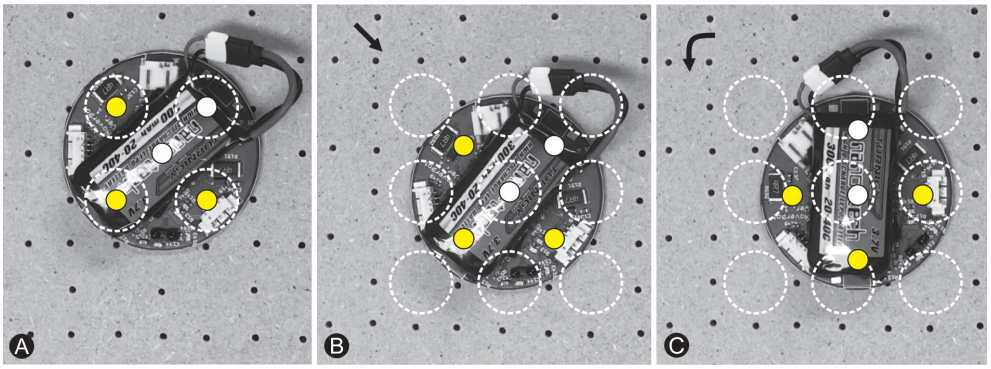


Figure 3.42: Recovery from locked position. A) HoverBot is locked in a 45 degree angled position. Four of its five coils are aligned with permanent magnets. B) HoverBot rapidly pulsed its center coil and regained center coil alignment with a permanent magnet. C) HoverBot additionally actuated a side coil and regained a slightly shifted idle position. At this stage, HoverBot is reenabled to move.

3.6.3 Formation, Random Movement, and Collision

Figure 3.43 shows four demonstrations with HoverBots version 1: A) moving in formation; B) moving randomly, colliding, and recovering from rotation; C) colliding; and D) colliding while one robot is in sleep mode acting as a passive agent. The following qrcode contains a link to the demonstrations.

Formation, movement, and collision

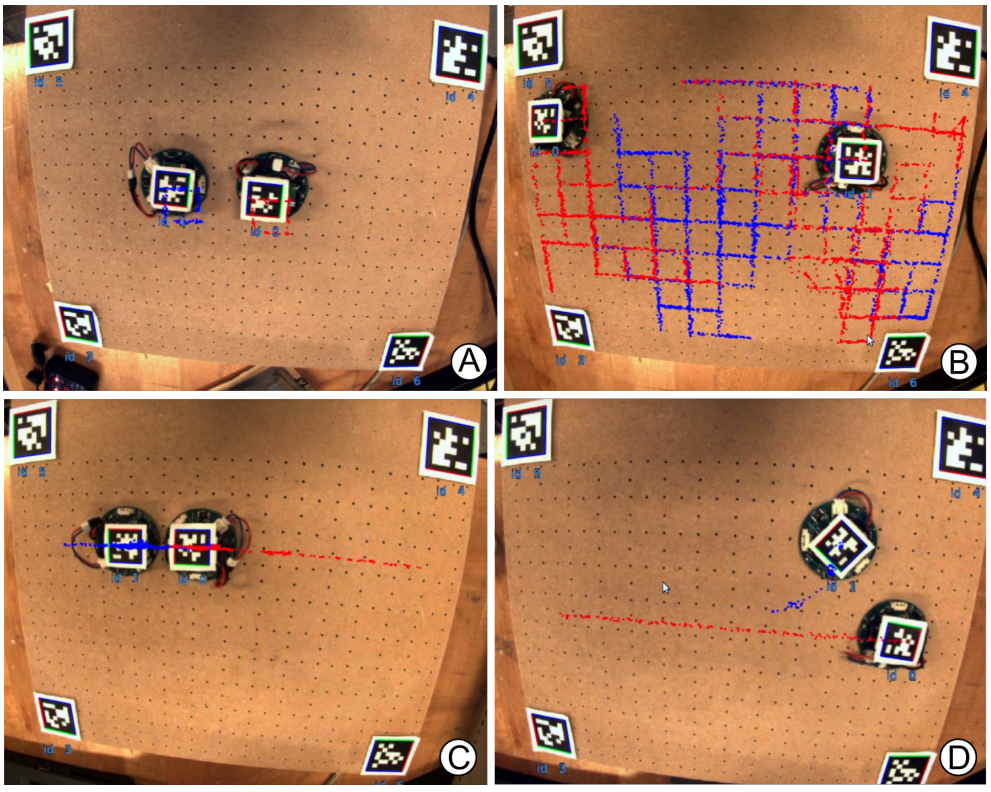


Figure 3.43: Demonstrations of the locomotion capabilities of multiple HoverBots version 1. A) Two HoverBots circle in formation until they are unsynchronized. B) Two HoverBots move randomly, collide, and recover. (C) Two HoverBots collide frontally with one another. (D) One HoverBot collides with a passive HoverBot. Red and blue trajectories depict HoverBot's movements over time.

Chapter 4

Multi-Functional Sensing Through Time Series Classification

In Chapter 2, I discuss the multi-functional sensing capabilities of previous swarm robotic systems. The vast majority of them are capable of multi functional sensing with ≥ 2 measurands. The work in this chapter empirically shows how HoverBot's single Hall-effect sensor can be augmented for the detection of three measurands: collision, rotation, and successful movement by modifying and applying signal processing techniques to magnetic field readings. I specify in Section 4.9 under which circumstances my techniques might be applied to other sensors, and I give a brief insight into the research opportunities that potentially arise from this approach more generally. This chapter is based on my published journal paper [7], which I put in Appendix F. The following qrcode contains a link to a video that summarises my work in this chapter.

Multi-functional sensing with HoverBots

4.1 HoverBot's Instrument Model

Figure 4.1 shows HoverBot's instrument model. The measurands successful movement, collision and rotation can be related to the physical measurement variable magnetic flux density. HoverBot uses a Hall-effect sensor to convert the magnetic flux densities into voltages which are subsequently converted via an analogue digital converter to digital measurements. The microcontroller processes the samples and checks them against a previously trained classifier. My classifier combines dynamic time warping and barycenter averaging to build time-variant representations of the measurands.

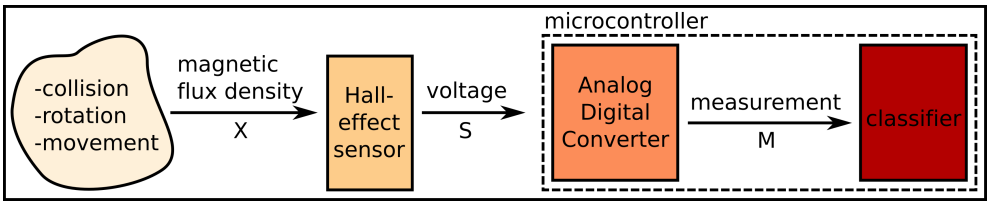


Figure 4.1: HoverBot detects collision, rotation and movement measurands by analysing magnetic flux densities. A Hall-effect sensor converts the magnetic field into a voltage which is then converted via an ADC to a digital signal. The classifier associates the measurements to one of the three measurands.

4.2 Magnetic Field Measurements

I simulated the magnetic field of the magnetic grid with FEMM, a simulator for solving low frequency electromagnetic problems on two-dimensional planar and axisymmetric domains [87]. Figure 4.3b shows the simulation results; the grey rectangles indicate magnets and the dashed lines serve as reference to the measurements in Figure 4.3c. I measured the magnetic field of the magnetic grid for verification using a HoverBot and its Hall-effect sensor. The simulated and measured magnetic fields broadly correlate with one another. The first amplitude in Figure 4.3c is slightly shifted due to fabrication tolerances. The pocket holes of the permanent magnets are slightly larger in diameter than the magnets themselves leading to imperfect magnet alignments.

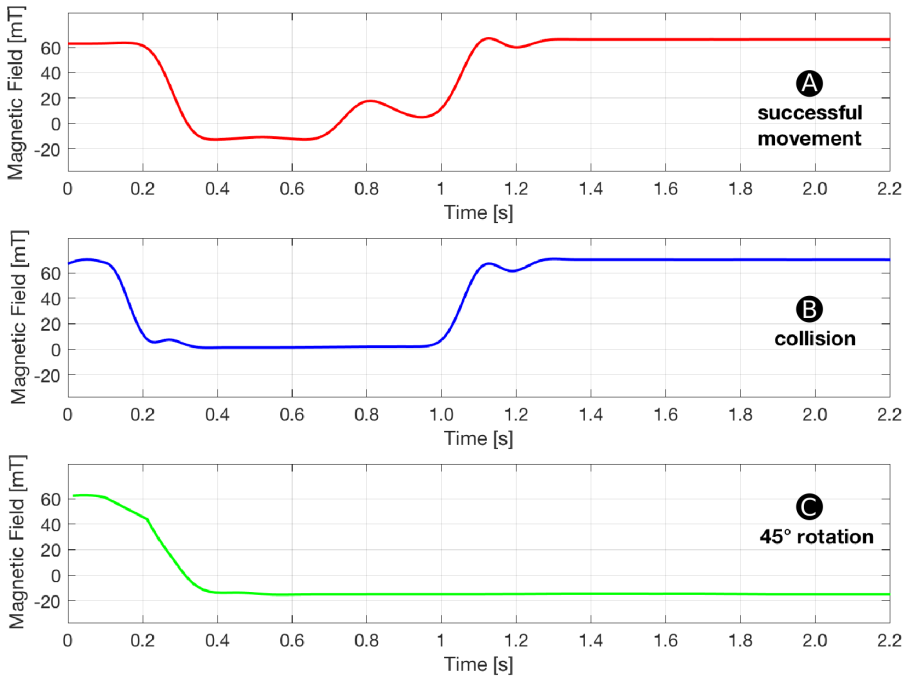


Figure 4.2: This figure shows examples of magnetic field measurements over time that a HoverBot measures during A) successful movement, B) collision, and C) 45-degree rotation. Each time series is distinctly different from the other.

HoverBot measures magnetic field values between -20 to $+60\mathrm{mT}$ as shown Figure 4.3c. However, the actual measurements during movement look somewhat different since they are not only position-dependent as in Figure 4.3c but also time-dependent. Figure 4.2 shows a set of time-dependent magnetic field measurements. Figure 4.2a indicates the magnetic field measurements of a HoverBot during a successful movement. When a HoverBot collides with a static object, its magnetic field measurements look distinctly different compared to its successful movement measurements as shown in 4.2b. Figure 4.2c indicates the magnetic field measurements that occur in a 45-degree rotated incident. The magnetic field profiles for successful movement and collision in Figures 4.2a and 4.2b possess distinct magnetic field changes over time. The magnetic field profile for rotation in Figure 4.2c differs from the other profiles since the magnetic field does not change anymore once the robot rotated and locked into position. Therefore, detecting rotation is a simple case of measuring a constant negative magnetic field over time. Since detecting rotation is trivial and does not add value to this work, I explicitly decide to focus on the remaining more challenging profiles.

One of HoverBot's advantages is its precise locomotion in Manhattan Geometry. Missteps or rotations are rare events. During data acquisition, I only observed successful movements and collisions. The robot randomly moved within the arena and occasionally collided with arena boarders. My dataset consists of 259 samples, whereas 203 samples are from successful movements and 56 samples are from collisions. Each sample is a collection of timestamps, magnetic field measurements, x-position, y-position and orientation of HoverBot. HoverBot transmitted its magnetic field measurements via infrared. I acquired the data with the experimental setup that I describe in Sections 3.4 and 3.5.

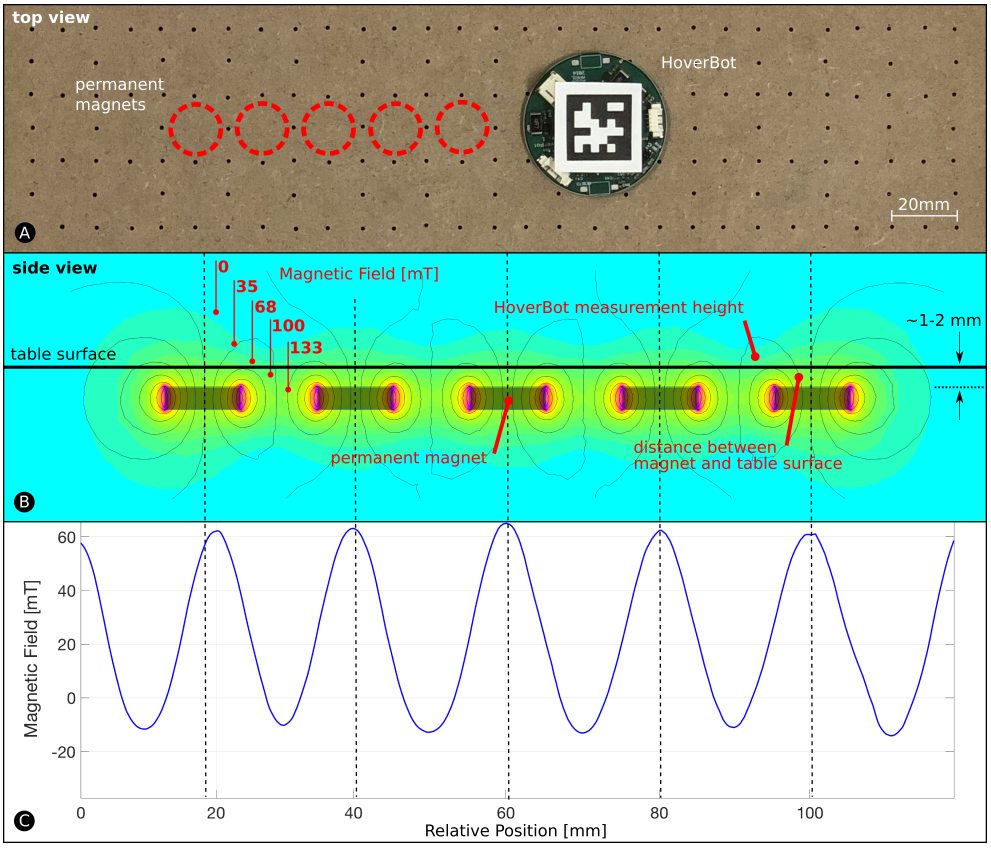


Figure 4.3: This figure compares the finite element method (simulation) of the magnetic field of five permanent magnets with actual measurements that I took with a HoverBot. A) HoverBot moved along five magnets. B) I simulated the magnetic field that HoverBot experienced during this movement. The permanent magnets are embedded approximately 1 millimetre beneath the top surface of the magnet-levitation table. C) Magnetic field measurements from HoverBot's Hall-effect sensor during movement. HoverBot approximately hovered $200\mu\text{m}$ above the surface during these measurements. Measurements and computer simulation broadly correlate.

4.3 Problem Definition

A major challenge in discriminating between successful movement and collision profiles is their variations in the time and measurement domain. For example, a HoverBot's speed may vary between actuations, resulting in measurement signals that are stretched or compressed. Additionally, environmental factors can cause measurement values to vary over time, such as when HoverBot's elevation above the table (and hence, its position in the magnetic field) varies due to air pressure fluctuations in the magnet-levitation table. To classify the measurement profiles depicted in Figures 4.2a and 4.2b, I must account for these variations while operating within the computational constraints of HoverBot's microcontroller.

I describe in the following sections a classification method that learns offline representations of each measurement profile and stores these representations on the robot. When the HoverBot agent obtains a new series of measurements online, the measurements are compared to the stored class representations to determine the maximum likelihood classification. My classification procedure builds on several component techniques from the field of signal processing. I introduce each of these components in turn in Section 4.4 and then incorporate them into my method in Sections 4.5 and 4.6. Throughout Section 4.4, I refer to Figures 4.4 and 4.5 for specific examples.

4.4 Classifier Training

4.4.1 Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) [88] is used to align two time sequences of potentially different length and measure the amount of similarity between them. DTW finds correspondences between points from the two sequences by warping them in the time domain. Given a distance measure, DTW computes the set of point correspondences that minimizes the cumulative distance between the sequences. Figure 4.4 shows an example of DTW for two arbitrary signals. Consider two input signals $x = [x_1, \ldots, x_M]$ and $y = [y_1, \ldots, y_N]$. Let $d(x_i, y_j)$ be a measure of the distance between x_i and y_j, such as the squared Euclidean distance $d(x_i, y_j) = (x_i - y_j)^2$. First, DTW computes the distance matrix d (Figure 4.4a) and subsequently the accumulated distance matrix D (Figure 4.4b). DTW initialises the first row and column of D as follows:

$$D_{1,1} = d(x_1, y_1) \tag{4.1}$$

$$D_{i,1} = D_{i-1,1} + d(x_i, y_1), \quad i = 2, ..., M \tag{4.2}$$

$$D_{1,j} = D_{1,j-1} + d(x_1, y_j), \quad j = 2, ..., N \tag{4.3}$$

The rest of D is then computed as:

$$D_{i,j} = d(x_i, y_j) + \min \begin{cases} D_{i-1,j} & i = 2, ..., M \\ D_{i,j-1} & j = 2, ..., N \\ D_{i-1,j-1} & \end{cases} \tag{4.4}$$

The DTW algorithm uses dynamic programming to build D, where each element $D_{i,j}$ gives the minimum cumulative distance between subsequences

$[x_1, \ldots, x_i]$ and $[y_1, \ldots, y_j]$. The final element of D, $D_{m,n}$, represents the minimum cumulative distance between the signals under any warping configuration, thus serving as a measure of similarity between the signals. Such an optimal warping configuration is given by the warping path $\phi^{x,y} = [(\phi_1^x, \phi_1^y), ..., (\phi_K^x, \phi_K^y)]$ through matrix D (see Figure 4.4c). Each element $\phi_k^{x,y} = (\phi_k^x, \phi_k^y)$ contains an index ϕ_k^x of signal x that has been matched to index ϕ_k^y of signal y. The warping path can be computed as:

$$\phi_K^{x,y} = (M, N) \tag{4.5}$$

$$\phi_k^{x,y} = arg\min_{i,j} D_{i,j} \quad i,j = \begin{cases} \phi_{k+1}^x - 1, \phi_{k+1}^y \\ \phi_{k+1}^x, \phi_{k+1}^y - 1 \\ \phi_{k+1}^x - 1, \phi_{k+1}^y - 1 \end{cases} \quad k = K-1, ..., 2 \tag{4.6}$$

$$\phi_1^{x,y} = (1, 1) \tag{4.7}$$

The length K of warping path $\phi^{x,y}$ varies depending on the extent to which the signals are warped, but it will satisfy the inequality $\max(M, N) \leq K \leq M + N - 1$.

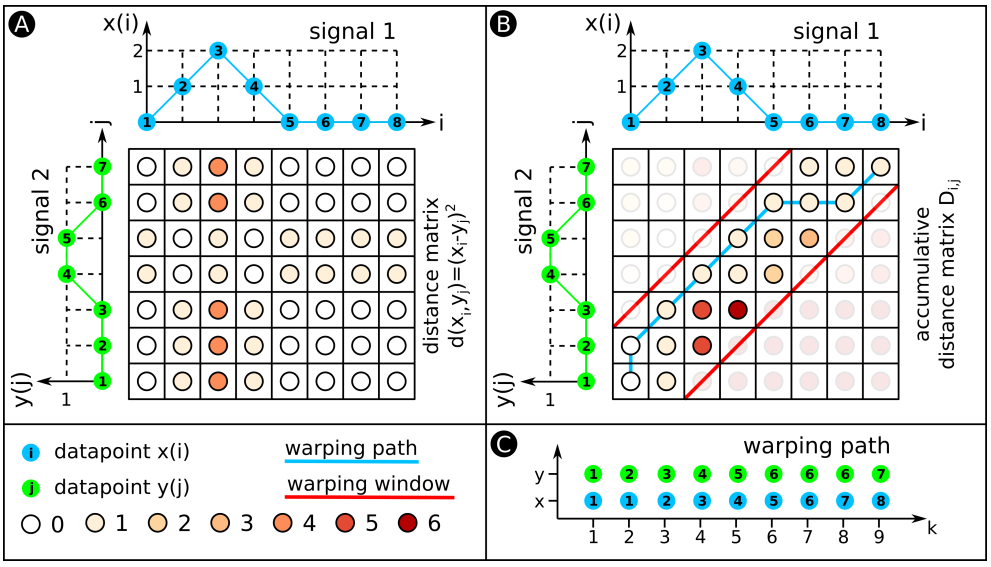


Figure 4.4: This figure exemplarily explains DTW. A) I compute a matrix in which each entry is the Euclidean distance between datapoints from signals 1 and 2. The colour code for the various distances is shown below. B) I use the distance matrix from (A) to develop the accumulated distance matrix D. The two red lines indicate the warping window. The warping path illustrated via the blue line has to fit within the borders of the warping window. C) The warping path explicitly states which datapoints of signal 1 align with what datapoints of signal 2. The warping path is always at least as long as the longest signal that is warped. In this example, the warping path exceeds both signals' length, the warping path is 9 datapoints long.

4.4.2 Constrained Dynamic Time Warping

The DTW formulation of the previous section is unconstrained i.e. the algorithm considers any possible warping configuration in matrix D. This can lead to pathological warping configurations wherein a single point from one signal is matched with many points from the other signal. Unconstrained DTW is thus sensitive to spurious alignments between signals. For example, unconstrained DTW might warp the collision time series in Figure 4.2b onto the successful movement time series in 4.2a even though the signals correspond to disparate events. Given these limitations of unconstrained DTW, it is common to limit the extent to which signals can be warped in the time domain. In Constrained Dynamic Time Warping (CDTW), we allow correspondences between points only if those points occur within a fixed time period of one another. The length of the warping window W is application-dependent and defined as $|n - m| \leq W$. W determines how many elements of matrices d (distance matrix) and D (accumulated distance matrix) are calculated. I show an example of a warping window in Figure 4.4b. CDTW can offer benefits in terms of time- and space-complexity, both are important for embedded platforms like the HoverBot. Because the warping path is constrained, only a portion of the matrix D must be computed and stored [88]. As a result, the time- and space-complexity are both reduced from $\mathcal{O}(N \times M)$ to $\mathcal{O}(N \times W)$, where the length of the warping window W is much less than M, the length of signal y.

4.4.3 DTW Barycenter Averaging (DBA)

For a classification task, it is often useful to compute a summary representation of a class of data. This averaging process is non-trivial when performed on variable-length signals. Naïve approaches based on pairwise alignment and averaging are sensitive to ordering effects and produce prohibitively long alignment sequences [89]. I utilise instead DTW Barycenter Averaging (DBA) to compute the average signal μ and the standard deviation σ of a group of signals [90]. Figure 4.5 shows an example of DBA for two arbitrary signals.

Consider a group of signals X to be input to the DBA algorithm. DBA initializes σ to be zero and randomly selects a signal $x^{(0)} = [x_1^{(0)}, ..., x_K^{(0)}]$ to serve as the initial average signal μ (Figure 4.5a) as

$$\mu_k = x_k^0, k = 1, ..., K \tag{4.8}$$

DBA also initializes sets $\psi_k = \emptyset, k = 1, ..., K$ that are used in the average computation. Each set ψ_k contains $z \in \psi_k$ elements for each specific $k \in K$, whereas z is a placeholder for aligned datapoints (Figure 4.5B). Each signal $x \in X \setminus x^{(0)}$ is aligned with the average signal using CDTW. Each set ψ_k is updated as

$$\psi_k = \psi_k \cup \{x_{\phi_i^x} : \phi_i^\mu = k\} \tag{4.9}$$

Please note that my example in Figure 4.5 only shows DBA for two signals. In Figure 4.5b, set ψ_1 consists of three aligned datapoints and sets ψ_{2-8} of two aligned datapoints. Once this process has been repeated for all signals, the value of each element of the average signal is updated as the barycenter of all points that map to the corresponding element of the existing average signal μ (Figure 4.5c); that is,

$$\mu_k = \frac{\sum_{z \in \psi_k} z}{|\psi_k|}. \tag{4.10}$$

Likewise, the elements of the standard deviation σ (Figure 4.5c) are updated as

$$\sigma_k = \sqrt{\frac{\sum_{z \in \psi_k} (z - \mu_k)^2}{|\psi_k|}} \tag{4.11}$$

The resulting updated average signal retains its original length while incorporating information from the entire group of signals. The process of aligning the group of signals with the average signal and computing an updated average and standard deviation can be repeated multiple times for better convergence.

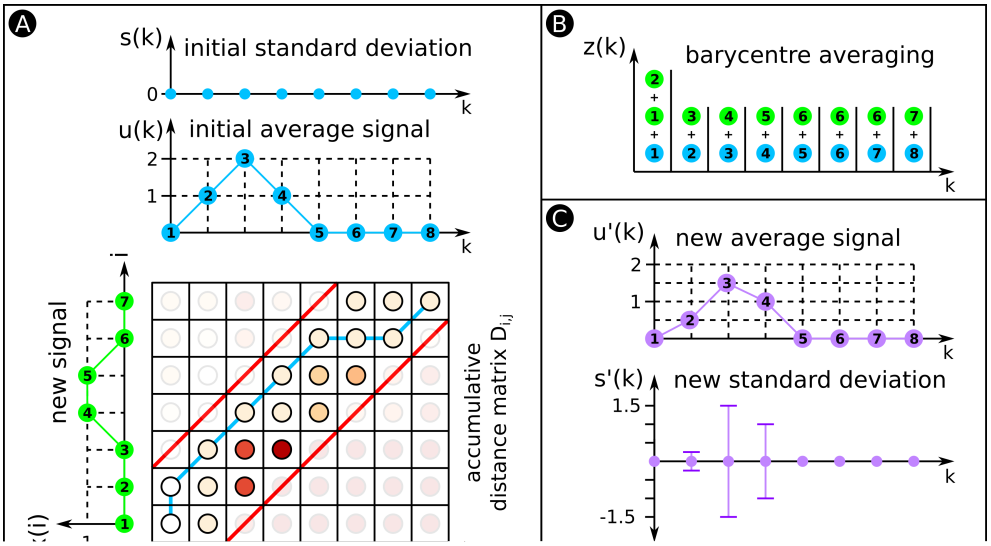


Figure 4.5: A) Similar to Figure 4.4b. This time the signals are initial average signal $u(k)$ and new signal $x(i)$, whereas $x(i)$ will be aligned and averaged with $u(k)$. B) DBA performs DTW to align both signals. The warping path indicates which datapoints from $x(i)$ are aligned with what datapoints from $u(k)$. There is a container for each k; I copy the datapoints that are warped to a specific k (see (a)) into the corresponding container. C) I compute from the datapoints of each container the average signal and standard deviation (SD). To build the average signal and SD of a group of signals, I warp all signals with the initial average signal. This process fills the various containers with many more datapoints, however, the technique remains the same.

4.4.4 Downsampling

My method can be implemented in several variants. Unconstrained with a complexity of $\mathcal{O}(N \times M)$, constrained with a complexity of $\mathcal{O}(N \times W)$ using a warping window, or even further constrained with a complexity of $\mathcal{O}(K \times W)$ through a combination of warping window and downsampling. Consider an input signal $x = [x_1, \ldots, x_N]$ with length N. I reduce the signal's length from N to K by dividing the signal x into K parts, each of which contains $L = \frac{N}{K}$ measurements. The value of each element $x_k^{'}$ of the downsampled signal $x^{'}$ is given by

$$x_k^{'} = \frac{1}{L} \sum_{i=L(k-1)}^{Lk} x_i, \quad k = 1, ..., K \tag{4.12}$$

Downsampling has an impact on the classifier's detection rate. The intuition is, the more you downsample the input signals, the worse the detection rate becomes. Downsampling potentially averages out important signal features that otherwise helped the classifier to discriminate between signals. From an engineering perspective, the more you downsample the input signals, the less memory is required to store values of the cumulative distance matrix. Memory reductions are very useful since it lowers the requirements for my low-cost microcontroller. Hence, there is a tradeoff between classification performance and memory utilization. I am able to achieve high classification performance with acceptable memory usage as shown in Section 4.7.

4.5 Offline Learning of Class Representations

Figure 4.6a gives an overview of the components that are involved in the offline learning of class representations. First, I conduct a random-walk HoverBot experiment and record the measurements (x-, y-position, orientation, magnetic field measurement, timestamp). I separate the data into approximately two second intervals, HoverBot's coil actuation scheme, to obtain a dataset consisting of many training examples. I manually label these examples based on HoverBot's movement over time. I also downsample each example (from $N \geq 1000$ to $K = 20$) in accordance with the computational limitations of HoverBot's microcontroller.

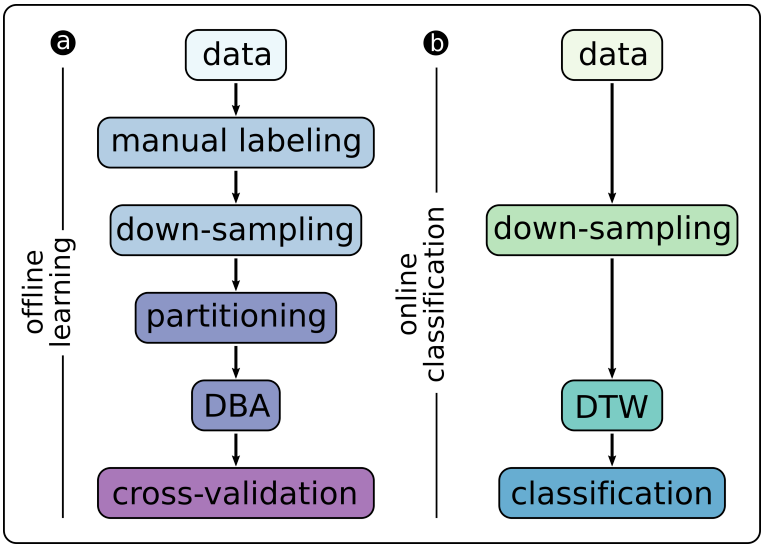


Figure 4.6: Overview of signal processing components for A) offline training of the classifier and B) online classification of new data samples.

Given this set of labelled training examples, I perform k-fold cross-validation ($KV = 10$). In each iteration, I divide (partition) the training data into labelled classes and perform DBA to obtain a representation of each class. In the validation step, I compute the Mahalanobis distance between each held-out example and the representations of each class:

$$d(\mu_k, \sigma_k, x_k) = \sqrt{\sum_{k=1}^{K} \frac{(x_k - \mu_k)^2}{\sigma_k}}. \tag{4.13}$$

The Mahalanobis distance measures the distance between a point and a distribution [91]. In other words, how many standard deviations σ is a point x away from the mean value μ. I classify each example according to the minimum-distance class, which corresponds to the maximum-likelihood classification.

4.6 Online Classification of Hall-Effect Measurements

Figure 4.6b indicates the components that are involved in the online classification. The components in Figure 4.6b are a subset of Figure 4.6a. HoverBot tries to move into a direction and records magnetic field measurements. Depending on the downsampling frequency, the HoverBot stores a number of average values into its memory ($K = 20$). The new dataset is used to calculate the Mahalanobis distance for each class representation (successful movement and collision). The Mahalanobis distances are compared with one another. If the distances do not reach a minimum value, the event will be labelled as unknown. Otherwise the event will be classified according to the lower Mahalanobis distance. The parameters of the class representations (μ_k, σ_k were trained offline, do not change during runtime and therefore are stored on the microcontroller's read-only-memory (ROM). The amount of ROM memory involved is dependent on the downsampling frequency.

4.7 Classifier Performance

Figure 4.7 shows the trained class representations for successful movement and collision. For each datapoint my classification method produces a mean value and a standard deviation. These values are constant and are stored in the microcontroller's flash memory.

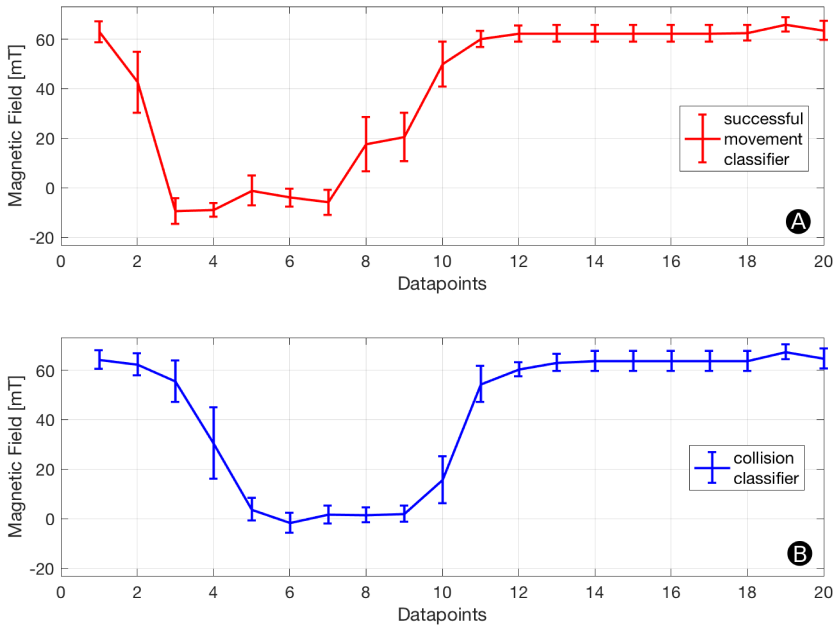


Figure 4.7: Each datapoint has a mean value and SD associated to it. These constants are stored in the microcontroller's ROM and used for online classification. The graphs indicate how much tolerance (SD) is permitted at each datapoint.

Figure 4.8 indicates the detection rates of my successful movement and collision classifier. For each number of datapoints (K), I compute a confusion matrix through k-fold cross-validation as described in Section 4.5. The detection rate for a successful movement is calculated by the True-Positive-Rate and the detection rate for a collision by the False-Positive-Rate of the confusion matrix. I show a confusion matrix example for $K = 20$ in Figure 4.8. While the detection rate increases with the number of datapoints, it starts stagnating once it exceeds 20 datapoints per sample. Therefore, I chose $K = 20$ in my setup achieving collision and successful movement detection rates of greater

than 85%. This setup is a fair trade-off between detection rate and computational effort. The microcontroller has to store $20 \times 20 \times 2$ bytes, assuming a 16-bit unsigned integer for each magnetic field measurement and not applying a warping window which otherwise decreased the memory requirement to $20 \times W \times 2$ bytes. HoverBot's microcontroller (Atmel SAMD21E16) comes with 8kB SRAM which comfortably meets the memory requirements of my classification method.

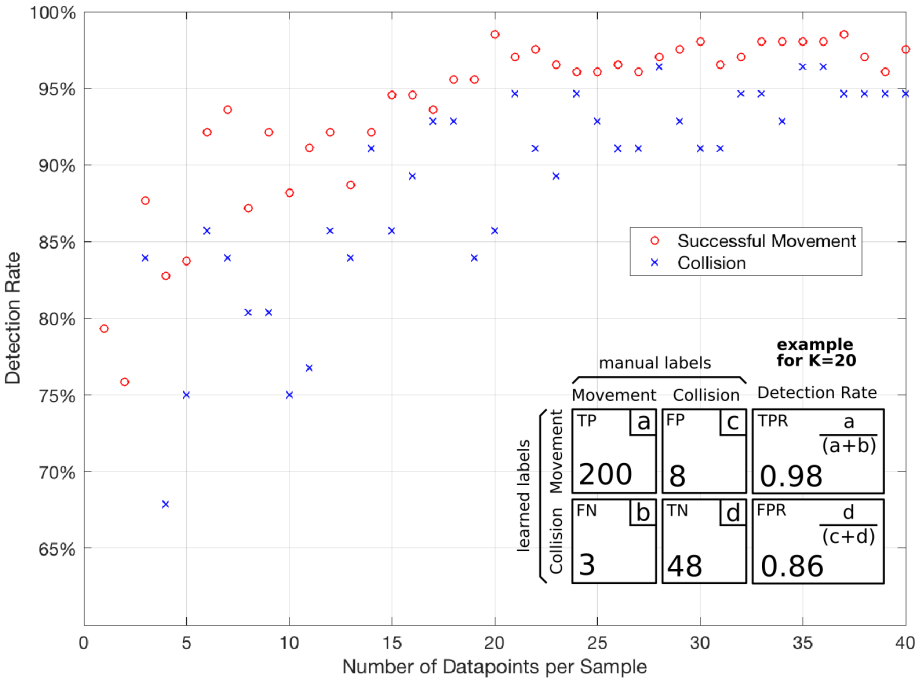


Figure 4.8: This graph shows the effect of downsampling on the classifier's detection rate. The detection rate exponentially increases until it starts stagnating once it exceeds 20 datapoints per sample. In the bottom right corner of this figure, I show an example of the confusion matrix for 20 datapoints. Confusion matrix legend: TP: True Positive, FN: False Negative, FP: False Positive, TN: True Negative, TPR: True Positive Ratio, FPR: False Positive Ratio.

4.8 Discussion

This work combines two rather unrelated research fields, the fields of signal processing and swarm robotics. The signal processing techniques were specifically adapted to operate on low-cost hardware. I introduced a warping window and a downsampling method to reduce the classifier's time- and space-complexity. I discuss the impact of downsampling on time varying signals in Section 4.8.1. I hope that this work encourages other swarm roboticists to reanalyse the sensing capabilities of their robots. The requirements and limitations of my approach are discussed in Sections 4.8.2 and 4.9. There might be an advantage of adding dedicated integrated circuits (IC) or Field Programmable Gate Arrays (FPGAs) to robot designs to process sensor data on the side to augment a robot's sensing capabilities. The advantages and disadvantages of specialized sensors are discussed in Section 4.8.3.

4.8.1 Downsampling

My downsampling method decreases a signal's length to a number of averaged datapoints. HoverBot's magnetic field profiles for successful movement and collision are simple. The profiles do not contain high frequency features, hence downsampling only had a limited impact on our classifier's detection rate. Classifiers built upon more complicated time-variant data are expected to be more heavily influenced by our downsampling method.

4.8.2 Limitations

HoverBot's discrete movement helped the discovery that signatures can be used to measure several measurands with a single sensor. If HoverBot's movement was continuous, I could still chop the measurements into the measurement profiles as shown in Figure 4.2, since any continuous movement can be regarded as a finite number of discrete movements with an infinitely small time difference in between them. However, the signatures are constrained to HoverBot's movement behaviour. HoverBot moves on a grid in Manhattan geometry which ensures that it measures a reoccurring magnetic field pattern. Other movement geometries such as continuous movements in arbitrary directions would impact the signatures and hence the time sequence classification.

4.8.3 Multi-Functional Sensing vs Specialised Sensors

The signatures allow the binary detection of successful movements, collisions, and rotations. These measurands can also be measured with specialised sensors. For example, I could detect collisions through tactile sensors that output continuous measurements. However, this is not a limitation of the time sequence classification but of the signatures. If I can find distinct signatures for each collision level, the time sequence classification will enable me to discriminate between them. The advantage of specialised sensors, however, is that they take away the processing behind the time sequence classification and enables continuous monitoring. For example, my approach first obtains a new data set and subsequently analyses it for collisions; I can only detect collisions once every movement cycle, whereas tactile sensors can detect collisions at any given time. A movement cycle is defined as a discrete movement from one permanent magnet to another. This might have an impact on the reactivity of robots. The disadvantage of specialised sensors is their cost.

4.9 Applicability to Other Systems

This study has demonstrated how time sequence classification can be used to measure several measurands with a single sensor. My method's applicability is dependent on signatures, unique measurement profiles that can be associated to specific measurands. Although I have not applied time sequence classification to other sensors or robots, I argue that it is generally applicable if the signature:

1. contains time varying measurements,
2. is systematically reoccurring, and
3. is distinctly different to other signatures.

4.9.1 Sensing Modalities

I believe that some sensing modalities are more suitable for time series classification than others. In general, we can discriminate between proprioceptive and exteroceptive sensing capabilities. Both terms owe their origins to biology;

the former one describes the sense of stimuli that are produced and perceived within an organism; the latter one describes the sense of stimuli that are external to an organism. For a better understanding, Table 4.1 contains all sensors that have been used by swarm robotic systems. I categorised the sensors into proprio- and exteroceptive sensors.

Proprioceptive Sensors	**Exteroceptive Sensors**
Wheel encoder	Infrared transceiver
Torque sensor	Humidity sensor
Accelerometer	Tactile sensor
Inclinometer	Microphone
Gyroscope	Temperature sensor
Voltagemeter	Light sensor
	Camera
	Radio
	Antenna
	Force sensor
	RFID reader
	Magnetometer

Table 4.1: Sensors used by previous swarm robotic systems. Sensors are divided into proprioceptive and exteroceptive sensors.

4.9.2 Artificial Signatures

Since exteroceptive sensors provide information about the environment, I can modify the environment to influence measurements. I call signatures that are manually created rather than naturally occurring, artificial signatures. This work here presents artificial signatures. I embedded permanent magnets into the arena surface to create collision, successful movement, and rotation signatures. Rather than trying to think about the measurands that I wanted to sense with HoverBot's Hall-effect sensor, I had to ask myself what its measurements look like over time at several occasions.

4.9.3 Natural Signatures

Proprioceptive sensors provide information about a robot's internal state. Popular examples are actuator, orientation or movement feedback (e.g. wheel

encoder, IMU, or accelerometer). Proprioceptive sensor readouts are mostly independent of environments and therefore platform independent. Since environmental modifications have little or no effect on proprioceptive sensor readings, I infer that I cannot influence their measurements either. I call signatures that are build from proprioceptive measurements, natural signatures. Proprioceptive measurements are more difficult to manipulate than exteroceptive measurements. Of course, there are exceptions such as detecting collisions with an accelerometer; there might be environmental modifications that could lead to different signatures e.g. by embedding springs into obstacles to influence their mechanical response during collision.

4.9.4 Augmenting Existing Sensors

Artificial signatures, such as the magnetic field profiles that I present in this thesis, are very specific to the robot system and not easily transferable. However, one can become creative and make their own environmental modifications and create signatures to exploit a robot's existing sensing capabilities. Finding or creating signatures requires a thorough understanding of the system at hand. Please find below a couple of examples.

An accelerometer seems to be an ideal candidate for time sequence classification. Acceleration continuously changes over time with robot movement. An accelerometer is a proprioceptive sensor; therefore, signatures occur naturally. Now, I need to take a look at the acceleration measurements during specific events and investigate whether I can build signatures that fulfil these three criteria: 1) contain time varying measurements; 2) are systematically reoccurring; and 3) are distinct to other signatures. Intuitively, the acceleration measurements should look different for collisions and successful movements. The acceleration measurements should also change in magnitude and frequency for movements on increasingly rough surfaces, potentially allowing terrain detections. I postulate that the accelerometer does not suffer from low signal-to-noise ratios.

A time of flight (TOF) ranging sensor determines the distance to objects by measuring the time it takes a light beam to travel to an object and back. I can implement a TOF sensor pointing upwards to measure the distance to objects above the robot. Since TOF sensors are exteroceptive sensors, I have to think about environmental modifications that help creating signatures. For

example, I could install a 3D surface above the robot. The surface could have periodically reoccurring patterns similar to the magnetic fields illustrated in Figure 4.3 depicting a 'reference map' at any given time. Any straight movement should result in a periodically reoccurring measurement pattern which could potentially allow to verify a movement, collision, and distance. While this setup is more difficult to establish than using the accelerometer, it uses a completely new dimension, the area above a robot, that could allow a robot to perform a variety of new tasks. The surfaces can be arbitrarily changed; each surface being a new robot environment with its own challenges. For example, a robot swarm could search for objects in the 'sky' investigating meta-heuristic search strategies. The interested reader is referred to Senanayake et al.'s literature review on search strategies for swarm robots [92].

4.10 Research Applications

4.10.1 Collision Dependent Swarm Behaviours

Rotation and successful movement detection are proprioceptive sensing capabilities, hence give insight into the internal state of a robot. Counting successful movements is useful for the robot to keep track of its position (odometry), and detecting rotations to derive knowledge about its orientation. Detecting collisions is an exteroceptive sensing skill. It provides the robot with information about its surroundings. I give a brief overview over the few swarm robotic studies that deal with collisions and indicate how they were utilised in Section 2.4.3.

Overall, collision is a promising candidate for research on and the design of scalable robot behaviours since incidence of collisions usually increases with increasing group sizes. Scalable refers to the ability of a swarm to perform well with different group sizes which I discuss in Section 2.1.2. Collisions have only been rarely studied in the swarm robotics context. Kernbach, Schmickl, and Mayaa et al.'s work depict excellent starting points for future work on collisions [39][63][61]; the HoverBot system depicts a suitable research platform since it embraces collisions and is capable of detecting them.

4.10.2 Collective Perception with Robot Swarms

Other interesting work that might profit from my approach is research on collective perception. Collective perception broadly refers to collectives that explore an environment and evaluate its features [93]. The work that I present here has the potential to enhance a robot's sensing capabilities without modifying its hardware, hence, could add to the list of observable features for collective perception. Notable literature on collective perception includes Khaluf's work on detecting and marking features e.g. of pollution areas [94], Kornienko et al.'s work on investigating which sensing and processing steps should be done individually or collectively for collective perception with robot swarms [46], Schmickl et al.'s work on hop-count and trophallaxis-inspired strategies (the exchange of fluids between insects for communication) to collectively perceive targets [63], Mermoud et al.'s work on aggregation based strategies to collectively perceive and destroy specific targets [95], and Tarapore et al.'s work on collective perception strategies inspired by the adaptive immune response to discriminate between dangerous and friendly cells [96].

Chapter 5

Collision Based Mapping Using HoverBots

Swarm robotic systems tend to avoid collisions (see Section 2.4). HoverBots are swarm robots that are capable of colliding due to their collision friendly locomotion strategy: low-friction locomotion (see Section 3.1). They solely consist of a populated PCB and a battery. PCBs are made of glass-reinforced epoxy laminate which makes HoverBot robust to physical interactions. I developed a time series classifier that allows HoverBots to detect collisions with static objects (see Chapter 4). In this chapter, I combine all building blocks by demonstrating how a group of four HoverBots collectively explores and maps an environment by bumping into their surroundings. This work is by no means a contribution to mapping, but a demonstration that proves that robot collisions can be used for useful tasks.

5.1 Experimental Setup

Figure 5.1 shows the experimental setup. The magnet-levitation table is enveloped by walls. I placed a static obstacle in the centre of the arena. Although I could have placed many more, smaller and indeed bigger objects, into the arena, the walls and the centre obstacle generate a symmetric environment. Robot observations are easy to relate. HoverBots start at their nest sites, that are the positions that they occupy in Figure 5.1a. Nests are landmarks that robots use to place their observations in relation. Figure 5.1b shows the *fused* occupancy grid map of the entire HoverBot group. The map is initialised with unknown state values; there is a 50% chance that a cell is occupied or empty assuming no prior knowledge of the environment.

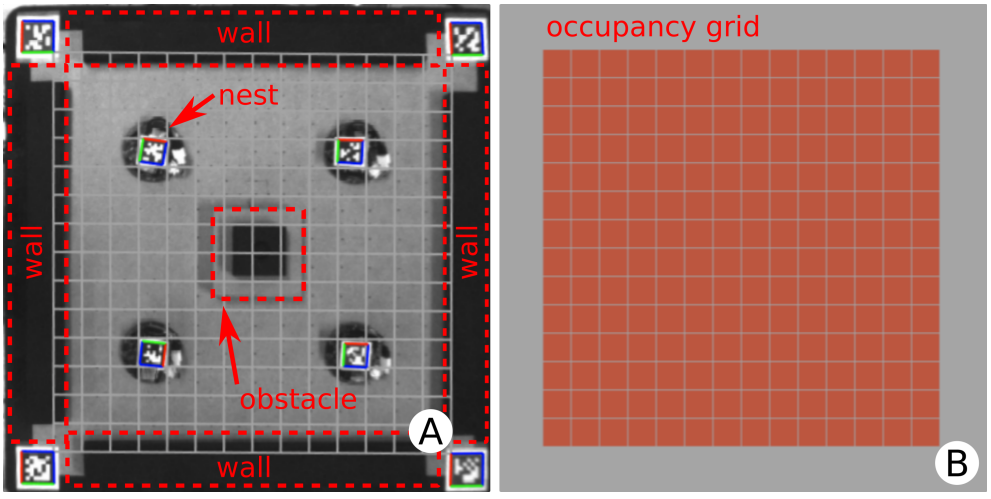


Figure 5.1: Experimental setup for collision mapping with HoverBots. A) HoverBots reside on their nest sites. There are four walls that constrain the arena. I placed an object into the centre of the robot arena to generate a symmetric robot environment. B) Fused observation matrix.

5.2 Explore-and-Return-to-Nest-after-N-Steps Collision Behaviour

Figure 5.2 illustrates the collision behaviour that HoverBots execute to map their environment. A robot starts at its nest and randomly explores its environment; I prohibit robots to make reverse movements within step size 1. For example, HoverBot can move *North* only if its previous movement was not *South*. HoverBots use a time sequence classifier to determine whether they moved successfully or collided after a step. The classifier helps the robot to make binary observations and to update its occupancy grid map. I explain the classifier in Chapter 4. A robot executes N steps, before it inverts the steps it has taken to return to its nest; the robot keeps track of all its movements $n \in N$. If a robot reaches its nest, the overhead camera notices and instructs the infrared transceiver to send a command to the robot. The camera tracks HoverBots through AprilTags. I explain the camera system in Section 3.4 and the infrared communication system in Section 3.5. If a robot returns to its nest and it received confirmation through the overhead infrared-camera system, it transmits the observations it has made during exploration via infrared to a PC. Once the observations are transmitted, it empties its memory and continues exploring. The fused occupancy grid map is the sum of observations from the four robots over a number of exploration walks. Every explore-and-return-to-nest adds another observation layer to the overall occupancy grid map. If a robot does not return successfully to its nest, it first initiates a systematic nest search. Starting from the believed nest position, it moves in increasing quadratic circles for $k \in K$ steps. If the systematic search fails too, the robot initiates a random walk to return to its nest by chance.

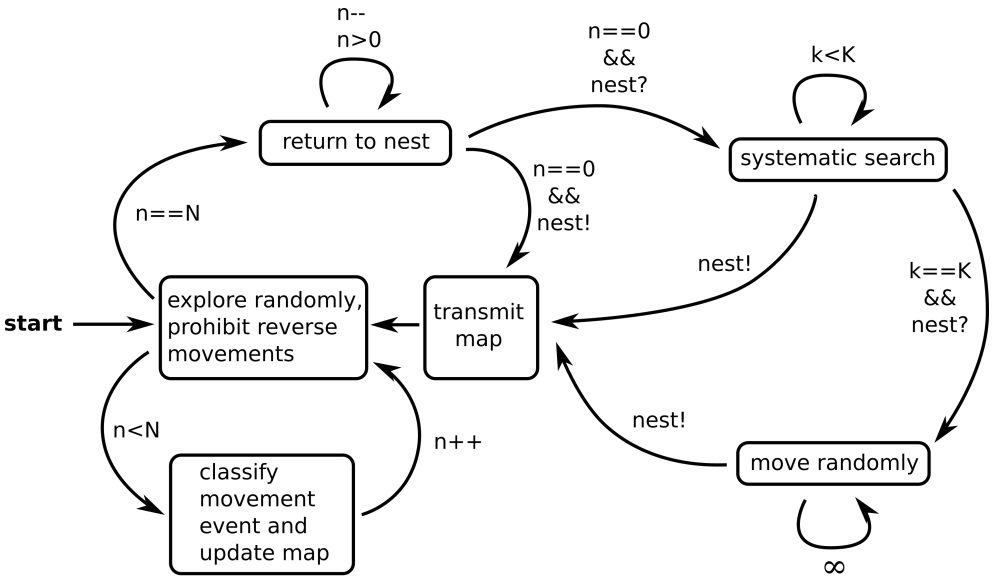


Figure 5.2: Robot collision behaviour to map an environment. A robot starts at its nest site. It performs a random walk for N steps before it returns home. During exploration phase $n \leq N$, HoverBots make observations by classifying their movements as successful movements or collisions. Once the robot returned to its nest, it transmits its observations via infrared to a centralised unit. If a robot does not make it back to the nest, it first initiates a systematic nest search, before it continues with a random nest search. An infrared transceiver and camera system are monitoring HoverBots. If a robot finds back to its nest, the system informs the robot.

5.3 What is N?

Figure 5.3 shows the fused observation matrix after each of the four robots executed several exploration walks, returned to their nests, and transmitted their observation matrices. The bottom left robot misclassified a cell, it returned an obstacle observation near its nest site although it is an empty space. If the robot makes another observation that indicates an empty cell where it previously detected an object, the fused observation matrix will cancel out these observations and the state of the cell becomes *unknown*.

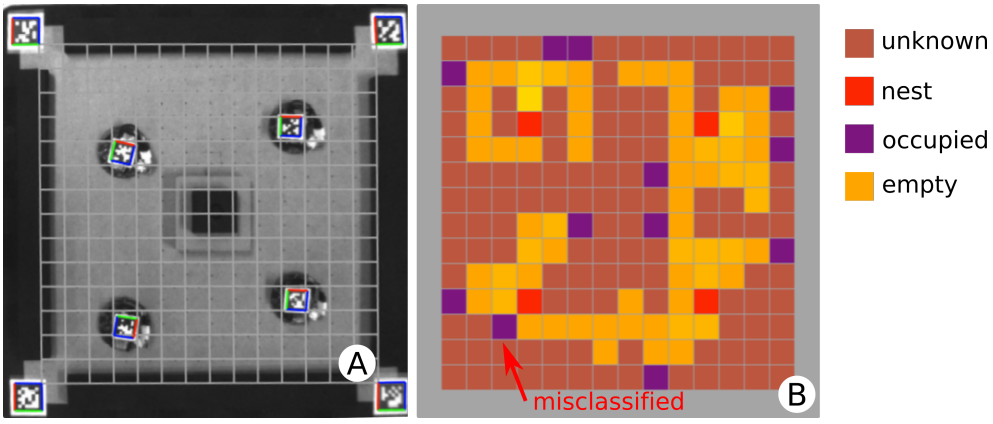


Figure 5.3: A) Four HoverBots map their environment by bumping into their surroundings and detecting collisions. B) The fused occupancy grid map of all observation matrices accumulated over time from four robots. The bottom left robot misclassified an empty space for an occupied cell. If the robot revisits the cell and makes more empty-cell classification, it will cancel its previous misclassification.

The more steps N a robot moves away from its nest, the more likely it is that it fails to return to its nest due to missteps. However, the larger N, the more cells the robot visits during exploration revealing larger parts of the map. Figure 5.4 shows the final occupancy grid maps of three experiments for: A) $N = 3$, B) $N = 4$, and C) $N = 5$ steps. The larger N, the more cells can be reached by a robot. Figure 5.4a shows a mine-sweeper like illustration. The numbers indicate the shortest distance from the nest site (red). Since robots move randomly, the probability to move in direction $d \in D$ with $D = [North, South, East, West]$ is $P(d) = \frac{1}{4}$. However, the probability to move North twice is: $P = \frac{1}{4} \times \frac{1}{3} = \frac{1}{12}$, hence less than 10%. Remember, once the robot moved into a direction, I prohibit it to reverse its previous step. I

mainly introduce this rule to make robots cover space and not only increase their knowledge about cells in close proximity to their nest. Therefore cells that are further away are less likely to be visited. Figure 5.4c shows the occupancy grid map with the most revealed cells. The walls and obstacle in the centre are identifiable. There are object detections between the top left and bottom left as well as between the bottom left and bottom right robots. These obstacle detections are due to robot-to-robot collisions.

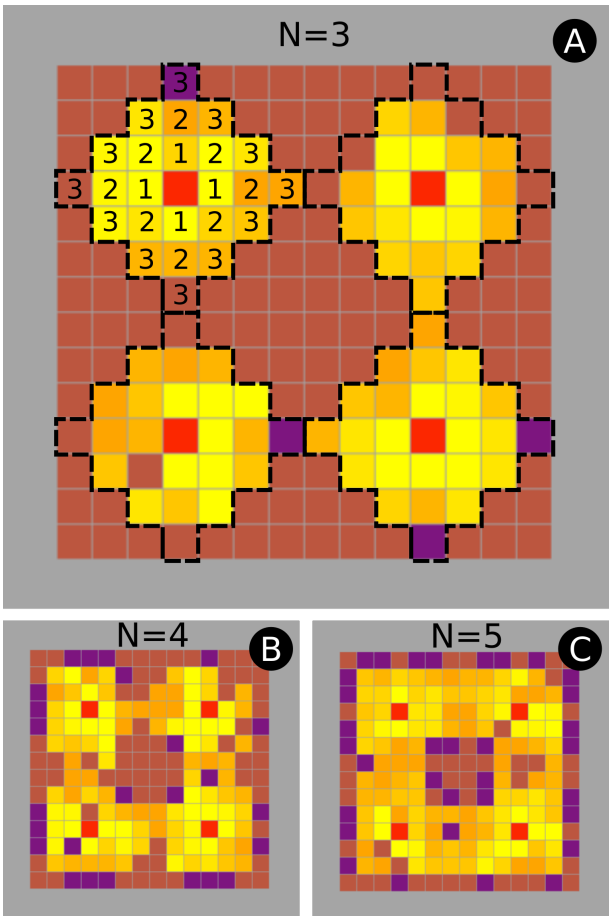


Figure 5.4: What is N? The more steps a robot is allowed to leave its nest for exploration, the more cells it uncovers. A) Robots are allowed to move $N = 3$ steps, B) $N = 4$ steps, C) $N = 5$ steps. The most complete occupancy grid map is shown in (C). There are some cells that cannot be reached by robots. Some cells are out of range N.

5.4 Occupancy Grid Map

Figure 5.5 shows the fused occupancy grid map of four robots over time. Robots are allowed to move $N = 5$ steps. The brighter the colour, the more often a robot made an observation for a certain cell. A robot explores N steps and moves back N steps to return to its nest. Once it is at its nest, it transmits its observation matrix to a PC. Therefore, in 20 steps, a robot generates two observation matrices, and four robots generate a total of 8 observation matrices. Figure 5.5f shows the final occupancy grid map. At this stage four robots moved each 170 steps; they stopped because their batteries depleted. Each robot managed to produce 17 observation matrices of its own quadrant. This map could now be used by other robots to localise themselves within the map. For example a robot could execute *north-east* movements until it reaches the top right corner of the map. From there the robot can reach any other position based on the map that has been created.

This demonstration is an example of mapping an environment through collisions. I developed this setup to prove a point: although most robots avoid collisions, collisions can be used for useful taks, here: mapping an environment. This work is the beginning of a much bigger research theme: collision robotics. I envision a subfield in robotics that solely focuses on how we can utilise collisions for useful robotic tasks. This might prove most interesting to swarm roboticists since collision avoidance is an expensive skill and collisions occur more often with increasing number of robots.

Collision mapping with HoverBots

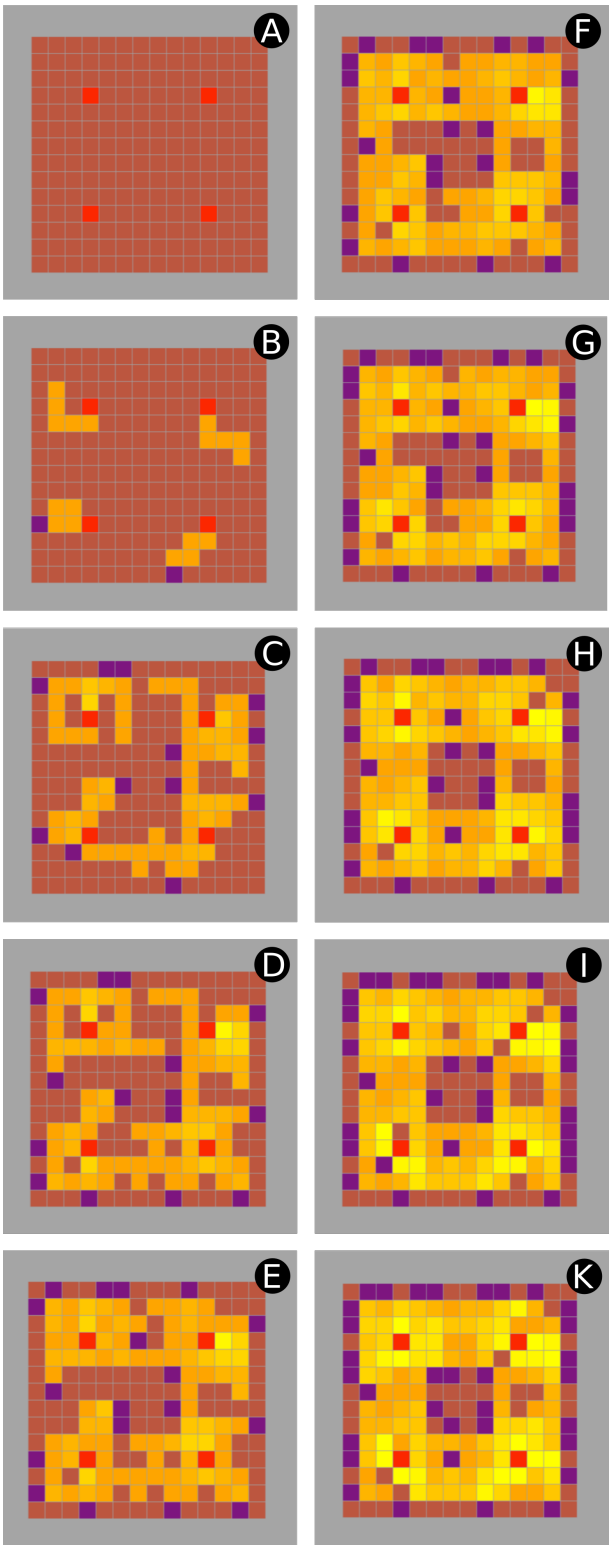


Figure 5.5: Four robots map their environment through collisions. The fused occupancy grid map is visualised in 20 step iterations. A) 0 steps, B) 10 steps, C) 30 steps, D) 50 steps, E) 70 steps, F) 90 steps, G) 110 steps, H) 130 steps, I) 150 steps, and K) 170 steps.

Chapter 6

Conclusion

6.1 HoverBots Are Easy to Manufacture

In this thesis, I invent and demonstrate a novel swarm robotic system that is easily manufacturable: the HoverBot system. In Chapter 3, I introduce the HoverBot system. HoverBots consist of a single PCB and a battery; they use low-friction locomotion. A magnet-levitation table supplies robots with upstreaming air and permanent magnets which I embedded into the table's top surface. I explain low-friction locomotion in Section 3.1: Robots' gravitational acceleration is encountered by an external force. This force can originate from a variety of technologies ranging from electro-static to buoyant levitation. Robots levitate which relaxes their actuator requirements. Much weaker forces can be used for robot locomotion, forces that would be insufficient if gravitational forces had not been encountered by an external force. I describe the magnet-levitation in Section 3.2. Robots are PCBs that possess electromagnetic actuators; they levitate on an air cushion; and interact with their planar coils with the permanent magnets. I describe HoverBots in Section 3.3. HoverBot is the easiest mass-manufacturable *autonomous* swarm robot that currently exists. Autonomous robots that only consist of a single populated PCB and a battery are difficult to outperform in terms of design for assembly (DFA) and design for manufacturing (DFM). Although the system is likely going to be improved, this thesis depicts the groundwork of the HoverBot system.

6.2 HoverBot's Functionality Can Be Enhanced Without Hardware Modifications

In Chapter 4, I develop a signal processing technique that enhances HoverBot's sensing capabilities without changing its hardware: I start by studying the capabilities of HoverBot's magnetic field sensor in Section 4.1. From there, I describe three different magnetic field profiles that HoverBot measures when it i) successfully moves; ii) collides with a static object; or iii) accidentally rotates by 45 degrees. In Section 4.3, I describe the challenges in discriminating between these magnetic field profiles. In Sections 4.4 to 4.6, I explain how I use many of such magnetic field profiles as data to train a time series classifier. Each profile class has its own representation which I upload on HoverBot's microcontroller. HoverBot's classifier takes a new measurement after each movement and analyses it for each profile class and assigns it to the maximum-likelihood class. I indicate the classifier's performance in Section 4.7. I discuss my signal processing approach in Section 4.8, its applicability to other sensors and robots in Section 4.9, and its research applications in Section 4.10.

This is an example of how signal processing techniques can be adapted for operation on low-cost microcontrollers. My work is a creative way of enhancing swarm robotic systems. It proves that there is still a lot of space for functionality enhancements without necessarily making hardware improvements. This technique or derivations of it might be applied to other sensors and robot systems. In any case, this technique will remain useful for roboticists who decide to conduct research with the HoverBot system or who implement active low-friction locomotion in new robots through aerodynamic levitation and magnetic field interactions.

6.3 HoverBots are Designed for Research on Robot Collision Behaviours

In Chapter 5, I start by exploring collision dependent swarm behaviours by taking advantage of HoverBots' capabilities to embrace and detect collisions: I let a group of four HoverBots collide with their surroundings to collectively map their environment. In Section 5.1, I give an overview of the experimental setup. I explain the mapping algorithm in Section 5.2; each robot is programmed with the same program with only slight alterations. I assign robots to nest sites and let them explore their environment for a certain number of steps before they have to return to share their observations with a centralised unit. The combination of observations emerges to a global map. In Section 5.3, I explain the impact of the number of robot steps on the overall mapping task. In Section 5.4, I show the end-product of the mapping task: an occupancy grid map. The occupancy grid map indicates the certainty that cells are empty or occupied. The more often a robot visits a specific cell, the more trustworthy are its observations.

This demonstration shows that robot collisions can be utilised for useful tasks. Collision avoidance does not only increase the cost of a robot due to specialised components that allow the detection and avoidance of collisions, but also prevent robots from using collisions to interact with other swarm robots and their environment.

Chapter 7

Future Work

This thesis builds the foundation of a lot of interesting work. In this chapter I map out some of my ideas that I came up with throughout my studies and which I urge others to continue to further this work.

7.1 Multilayer Planar Coils

In Section 3.3.4, I discuss the design parameters that influence the magnetic field of HoverBot's planar coils. While I optimised the planar coils by changing the trace width and spacing between traces according to the minimum specifications of the PCB foundry, it is also possible to embed planar coils in several PCB layers. In this case, it is important to get the stacking right. The magnetic field of each layer must align, hence the current direction has to be the same. This approach has several advantages such as: a coil with more windings and lower current induces the same magnetic field than a coil with fewer windings but larger current. I did not follow this design path since my HoverBot designs created sufficient magnetic forces for movement and I used the extra layers to make the robot as compact as possible.

7.2 Changing Magnet-Coil Pattern and Alternating Magnet Polarities

The magnetic grid pattern can be changed to other geometries such as to a radial pattern. A pattern change would have an influence on how the robot moves. In this case, the interactions between coils and magnets must be re-evaluated; the coil array geometry of the robot likely changes too. The advantage could be that robots move in more interesting trajectories than Manhattan geometry potentially allowing diagonal movements too (radial formation). It is also possible to develop a magnetic grid in which adjacent magnets possess opposing magnetic fields. In that case, the coil-to-coil distance can match the magnet-to-magnet distance, and coils can be operated like a stepper motor. This could lead to smooth robot movement.

7.3 Substituting Magnets with Electro-Magnetic Devices

The permanent magnets in the magnet-levitation table serve as magnetic anchors and energy fields that can be used to detect collisions, rotations, and successful movements. It is possible to substitute a few or all magnets with electro-magnetic elements or electro-permanent magnets to be able to make changes to the magnetic field of the magnet-levitation table. This alternation would allow interesting experiments in which HoverBots look for specific magnetic fields. It would also be possible to let robots control electro-magnetic devices. I explain this idea in more detail in Section 7.5. Permanent magnets could be also replaced by ferro-magnetic primer. Ferro-magnetic primer is a type of paint that can be applied to substrates. The interactions between HoverBot coils and primer might be sufficient to achieve locomotion. Magnetic primer can also be applied to HoverBots. Figure 7.1 shows a HoverBot with a magnetic primer coating on its bottom surface. The coating interacts with the permanent magnets and could help stabilise movement (prevent accidental rotations). It is also possible to paint passive objects such as wood discs with magnetic primer. Such discs could become interesting objects for HoverBots to interact with.

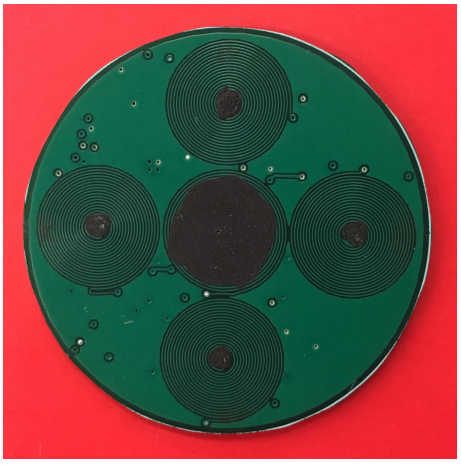


Figure 7.1: Ferro-magnetic primer can be used to paint key-elements of the planar coils. The primer can be useful to enhance interactions between coils and magnets.

7.4 Equipping Robots with Miniature Air-Blowers

Murata sells miniaturised microblowers that use ceramics as the driving source. The device utilises ultrasonic piezoelectric vibration of ceramics to generate high air pressure. The following qrcode shows an acrylic disc onto which I mounted a microblower. The microblower is driven by a small circuit that can easily be incorporated into the existing HoverBot circuitry. The advantage of an internal air source is that we do not need an air-blower anymore to make HoverBots levitate. Robots would have their own air-source which they can switch on and off whenever it suits them. The disadvantage is increased cost and another component that can potentially malfunction impacting robot mass-manufacture and robot maintenance.

Video: Murata airblower as air-source.

7.5 Electro-Magnetic Stigmergy

If we substitute permanent magnets with electro-magnetic elements such as coils, it is possible to build an electronic circuit that interfaces coils and an infrared transceiver. If a robot is above a coil, it can command via infrared to change the current flow through the coil. This setup gives the robot the opportunity to make changes to its environment which can be picked up by other robots. In nature this way of passive communication, the communication through environmental modifications, is called stigmergy [97].

7.6 Collective Robot Charging

Another option is to cover the entire magnet-levitation table with contact pads as shown in Figure 7.2. Robots do not touch the pads if the air source is switched on. An electronics circuit could interface the control of the air-source with the overhead infrared transceiver. Robots come to a collective agreement when it is time for charging and switch-off the air source via infrared. Robots make contact with the pads and charge. The following qrcode shows a proof of concept video. I put copper traces beneath an acrylic disc and connected them to a LED. When I switch off the air-source, the disk makes contact with the contact pads and the LED lights up.

Video: Charging mechanism.

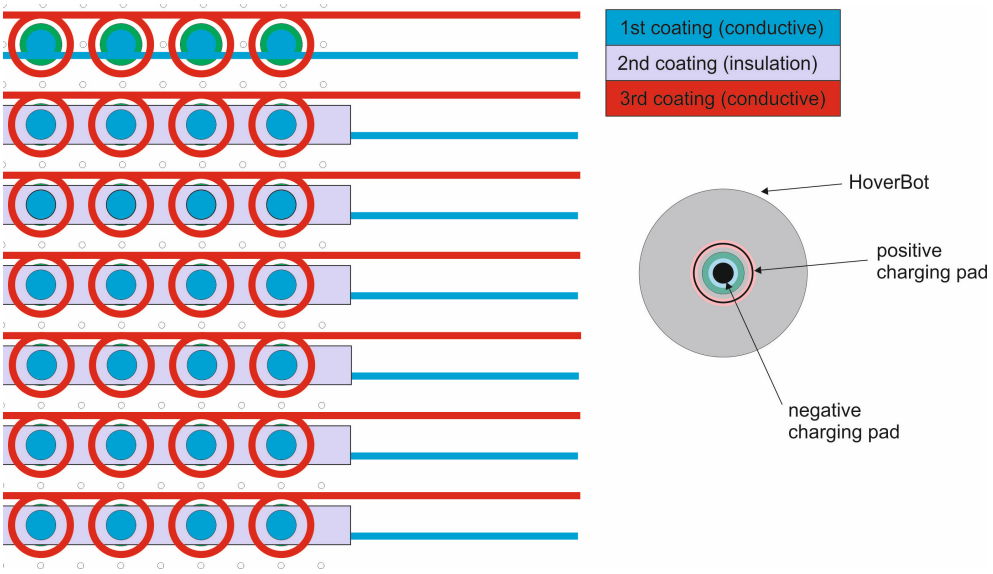


Figure 7.2: Charging mechanism for floating electronics.

7.7 Closed-Loop Control Using the Hall-Effect Sensor

HoverBot's control is currently open-loop. I energise the coils by programming conservative time-current profiles that lead to movement. The next improvement could be a closed-loop controller. Figure 7.3 shows the control diagram of a closed loop controller for HoverBot. Figure 7.4 shows the magnetic field profile and indicates the reference variables that could be used for a closed-loop controller. The controller would try to minimise the error $e = w - x_r$. I would start with a standard implementation of a Proportional Integral Derivative (PID) controller.

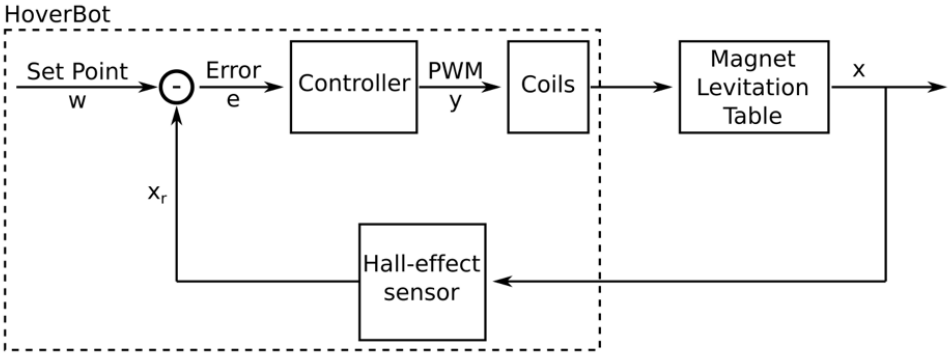


Figure 7.3: Closed-loop control scheme.

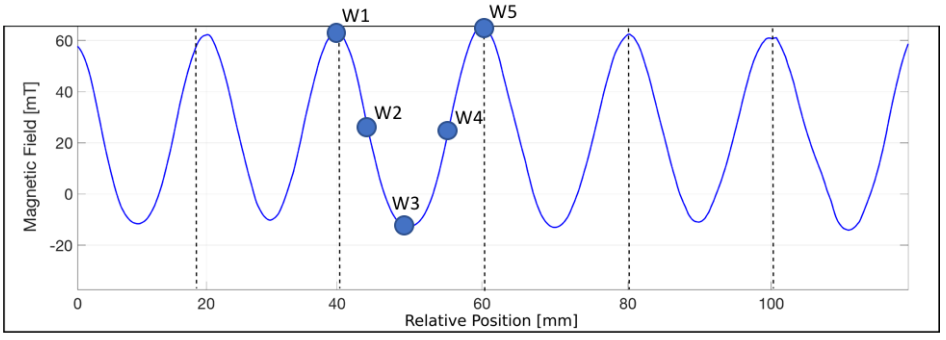


Figure 7.4: Magnetic field readings that can potentially be used for closed-loop control.

7.8 Collision Triggered Search

HoverBot version 2 possesses a proximity sensor that points upwards. This sensor can be used to measure the distance to any object above the robot. We can mount a structured layer (terrain) above the magnet-levitation table and let robots execute collision dependent search tasks. For example, robots could search for the global minimum distance in the terrain. Every time they collide, they perform a new distance measurement. Larger robot groups should find the global minimum faster. Terrains can be arbitrarily designed and exchanged which makes it a versatile experimental setup.

7.9 Using HoverTag as Sensor Node

The HoverTag is a multi-functional sensing node. Although I designed HoverTag to explore various sensors and their usefulness in future HoverBot projects, it is a fully functional device that is ready to be used in other projects. The HoverTag design was just recently used in the ORCA Hub, a British initiative to automate offshore plants. It has been used for the real time monitoring of critical offshore related processes. The HoverTag underwent some electrical modifications and has been renamed to Limpet underlying its new capability to be attached to nearly any surface.

References

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

[2] H. Hamann, *Swarm Robotics: A Formal Approach*. Springer, 2018.

[3] E. Bonabeau, D. d. R. D. F. Marco, M. Dorigo, G. Théraulaz, G. Theraulaz, *et al.*, *Swarm intelligence: from natural to artificial systems*. No. 1, Oxford university press, 1999.

[4] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014.

[5] M. P. Nemitz, M. E. Sayed, J. Mamish, G. Ferrer, L. Teng, R. McKenzie, A. O. Hero, E. Olson, and A. A. Stokes, "Hoverbots: Precise locomotion using robots that are designed for manufacturability," *Frontiers in Robotics and AI*, vol. 4, p. 55, 2017.

[6] M. P. Nemitz and A. A. Stokes, "A locomotion platform and multiagent system," *Patent application*, vol. GB1611448.0, 2016.

[7] M. P. Nemitz, R. Marcotte, M. E. Sayed, G. Ferrer, A. O. Hero, E. Olson, and A. A. Stokes, "Multi-functional sensing for swarm robots using time sequence classification: Hoverbot, an example," *Frontiers in Robotics and AI*, 2018.

[8] M. P. Nemitz and A. A. Stokes, "Hoverbots: Embracing and detecting collisions using robots designed for manufacturability," *International Conference on Robotics and Automation*, 2018.

[9] D. Ross, M. P. Nemitz, and A. A. Stokes, "Controlling and simulating soft robotic systems: insights from a thermodynamic perspective," *Soft Robotics*, vol. 3, no. 4, pp. 170–176, 2016.

[10] M. P. Nemitz, P. Mihayalov, T. W. Barraclough, D. Ross, and A. A. Stokes, "Using voice coils to actuate modular soft robots: Wormbot, an example," *Soft Robotics*, 2016.

[11] A. C. McConnell, M. Vallejo, R. C. Moioli, F. L. Brasil, N. Secciani, M. P. Nemitz, C. P. Riquart, D. W. Corne, P. A. Vargas, and A. A. Stokes, "sophia: soft orthotic physiotherapy hand interactive aid," *Frontiers in Mechanical Engineering*, vol. 3, p. 3, 2017.

[12] L. Teng, K. Jeronimo, T. Wei, M. P. Nemitz, G. Lyu, and A. A. Stokes, "Integrating soft sensor systems using conductive thread," *Journal of Micromechanics and Microengineering*, vol. 28, no. 5, p. 054001, 2018.

[13] S. T. Mahon, J. O. Roberts, M. E. Sayed, D. H.-T. Chun, S. Aracri, R. M. McKenzie, M. P. Nemitz, and A. A. Stokes, "Capability by stacking: The current design heuristic for soft robots,"

[14] B. J. Crespi and D. Yanega, "The definition of eusociality," *Behavioral Ecology*, vol. 6, no. 1, pp. 109–115, 1995.

[15] D. Cassill, W. R. Tschinkel, and S. B. Vinson, "Nest complexity, group size and brood rearing in the fire ant, solenopsis invicta," *Insectes Sociaux*, vol. 49, no. 2, pp. 158–163, 2002.

[16] C. R. Reid, M. J. Lutz, S. Powell, A. B. Kao, I. D. Couzin, and S. Garnier, "Army ants dynamically adjust living bridges in response to a cost–benefit trade-off," *Proceedings of the National Academy of Sciences*, vol. 112, no. 49, pp. 15113–15118, 2015.

[17] M. Mitchell, *Complexity: A guided tour*. Oxford University Press, 2009.

[18] D. R. Hofstadter, *Gödel, escher, bach*. Vintage Books New York, 1980.

[19] D. D. Chaplin, "Overview of the immune response," *Journal of Allergy and Clinical Immunology*, vol. 125, no. 2, pp. S3–S23, 2010.

[20] D. Merkle and C. Blum, "Swarm intelligence: Introduction and application," 2008.

[21] M. Dorigo, V. Trianni, E. Şahin, R. Groß, T. H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, *et al.*, "Evolving self-organizing behaviors for a swarm-bot," *Autonomous Robots*, vol. 17, no. 2-3, pp. 223–245, 2004.

[22] O. Soysal and E. Şahin, "A macroscopic model for self-organized aggregation in swarm robotic systems," in *International Workshop on Swarm Robotics*, pp. 27–42, Springer, 2006.

[23] J. McLurkin and J. Smith, "Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots," in *7th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, Citeseer, 2004.

[24] D. Payton, M. Daily, R. Estowski, M. Howard, and C. Lee, "Pheromone robotics," *Autonomous Robots*, vol. 11, no. 3, pp. 319–324, 2001.

[25] D. Payton, R. Estkowski, and M. Howard, "Pheromone robotics and the logic of virtual pheromones," in *International Workshop on Swarm Robotics*, pp. 45–57, Springer, 2004.

[26] M. J. Krieger, J.-B. Billeter, and L. Keller, "Ant-like task allocation and recruitment in cooperative robots," *Nature*, vol. 406, no. 6799, p. 992, 2000.

[27] W. Liu, A. Winfield, J. Sa, J. Chen, and L. Dou, "Strategies for energy optimisation in a swarm of foraging robots," in *International Workshop on Swarm Robotics*, pp. 14–26, Springer, 2006.

[28] K. Sugawara and M. Sano, "Cooperative acceleration of task performance: Foraging behavior of interacting multi-robots system," *Physica D: Nonlinear Phenomena*, vol. 100, no. 3-4, pp. 343–354, 1997.

[29] A. L. Christensen, R. O'Grady, and M. Dorigo, "A mechanism to self-assemble patterns with autonomous robots," in *European Conference on Artificial Life*, pp. 716–725, Springer, 2007.

[30] R. O'Grady, R. Groß, A. L. Christensen, F. Mondada, M. Bonani, and M. Dorigo, "Performance benefits of self-assembly in a swarm-bot," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 2381–2387, IEEE, 2007.

[31] N. R. Franks and T. Richardson, "Teaching in tandem-running ants," *Nature*, vol. 439, no. 7073, p. 153, 2006.

[32] V. Trianni and M. Dorigo, "Emergent collective decisions in a swarm of robots," in *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, pp. 241–248, IEEE, 2005.

[33] V. Trianni, S. Nolfi, and M. Dorigo, "Cooperative hole avoidance in a swarm-bot," *Robotics and Autonomous Systems*, vol. 54, no. 2, pp. 97–103, 2006.

[34] K. H. Petersen, *Collective construction by termite-inspired robots*. PhD thesis, 2014.

[35] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering," in *Proceedings of the 9th conference on autonomous robot systems and competitions*, vol. 1, pp. 59–65, IPCB: Instituto Politécnico de Castelo Branco, 2009.

[36] K. Gilpin, A. Knaian, and D. Rus, "Robot pebbles: One centimeter modules for programmable matter through self-disassembly," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 2485–2492, IEEE, 2010.

[37] M. W. Jorgensen, E. H. Ostergaard, and H. H. Lund, "Modular atron: Modules for a self-reconfigurable robot," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 2, pp. 2068–2073, IEEE, 2004.

[38] G. Caprari, P. Balmer, R. Piguet, and R. Siegwart, "The autonomous micro robot" alice": a platform for scientific and commercial applications," in *Micromechatronics and Human Science, 1998. MHS'98. Proceedings of the 1998 International Symposium on*, pp. 231–235, IEEE, 1998.

[39] S. Kernbach, R. Thenius, O. Kernbach, and T. Schmickl, "Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system," *Adaptive Behavior*, vol. 17, no. 3, pp. 237–259, 2009.

[40] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *Robotics and Automation*

(ICRA), 2012 IEEE International Conference on, pp. 3293–3298, IEEE, 2012.

[41] F. Mondada, E. Franzi, and P. Ienne, "Mobile robot miniaturisation: A tool for investigation in control algorithms," in *Experimental robotics III*, pp. 501–513, Springer, 1994.

[42] D. Pickem, M. Lee, and M. Egerstedt, "The gritsbot in its natural habitat-a multi-robot testbed," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 4062–4067, IEEE, 2015.

[43] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The robotarium: A remotely accessible swarm robotics research testbed," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 1699–1706, IEEE, 2017.

[44] G. Caprari and R. Siegwart, "Design and control of the mobile micro robot alice," in *Proceedings of the 2nd International Symposium on Autonomous Minirobots for Research and Edutainment, AMiRE 2003: 18-20 February 2003, Brisbane, Australia*, pp. 23–32, CITI, 2003.

[45] F. Mondada, A. Guignard, M. Bonani, D. Bar, M. Lauria, and D. Floreano, "Swarm-bot: From concept to implementation," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 2, pp. 1626–1631, IEEE, 2003.

[46] S. Kornienko, O. Kornienko, and P. Levi, "Minimalistic approach towards communication and perception in microrobotic swarms," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 2228–2234, IEEE, 2005.

[47] A. E. Turgut, F. Gokce, H. Celikkanat, L. Bayindir, and E. Sahin, "Kobot: A mobile robot designed specifically for swarm robotics research," *Middle East Technical University, Ankara, Turkey, METU-CENG-TR Tech. Rep*, vol. 5, no. 2007, 2007.

[48] M. Bonani, V. Longchamp, S. Magnenat, P. Rétornaz, D. Burnier, G. Roulet, F. Vaussard, H. Bleuler, and F. Mondada, "The marxbot, a miniature mobile robot opening new perspectives for the collective-robotic research," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 4187–4193, IEEE, 2010.

[49] F. Riedo, M. Chevalier, S. Magnenat, and F. Mondada, "Thymio ii, a robot that grows wiser with children," in *Advanced Robotics and its Social Impacts (ARSO), 2013 IEEE Workshop on*, pp. 187–193, IEEE, 2013.

[50] N. Farrow, J. Klingner, D. Reishus, and N. Correll, "Miniature six-channel range and bearing system: algorithm, analysis and experimental validation," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 6180–6185, IEEE, 2014.

[51] J. Seyfried, M. Szymanski, N. Bender, R. Estaña, M. Thiel, and H. Wörn, "The i-swarm project: Intelligent small world autonomous robots for micro-manipulation," in *International Workshop on Swarm Robotics*, pp. 70–83, Springer, 2004.

[52] J. McLurkin, A. J. Lynch, S. Rixner, T. W. Barr, A. Chou, K. Foster, and S. Bilstein, "A low-cost multi-robot system for research, teaching, and outreach," in *Distributed Autonomous Robotic Systems*, pp. 597–609, Springer, 2013.

[53] GCtronic, "Elisa-3," 2018. Available at `http://www.gctronic.com/doc/index.php/Elisa-3`.

[54] S. Wilson, R. Gameros, M. Sheely, M. Lin, K. Dover, R. Gevorkyan, M. Haberland, A. Bertozzi, and S. Berman, "Pheeno, a versatile swarm robotic research and education platform," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 884–891, 2016.

[55] P. Vartholomeos and E. Papadopoulos, "Analysis, design and control of a planar micro-robot driven by two centripetal-force actuators," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 649–654, IEEE, 2006.

[56] R. Groß, S. Magnenat, L. Küchler, V. Massaras, M. Bonani, and F. Mondada, "Towards an autonomous evolution of non-biological physical organisms," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5777 LNAI, no. PART 1, pp. 173–180, 2011.

[57] N. Napp, S. Burden, and E. Klavins, "Setpoint regulation for stochastically interacting robots," *Autonomous Robots*, vol. 30, no. 1, pp. 57–71, 2011.

[58] D. Cappelleri, D. Efthymiou, A. Goswami, N. Vitoroulis, and M. Zavlanos, "Towards mobile microrobot swarms for additive micromanufacturing," *International Journal of Advanced Robotic Systems*, vol. 11, no. 9, p. 150, 2014.

[59] R. Pelrine, A. Hsu, C. Cowan, and A. Wong-Foy, "Multi-agent systems using diamagnetic micro manipulation - from floating swarms to mobile sensors," in *Manipulation, Automation and Robotics at Small Scales (MARSS), 2017 International Conference on*, pp. 1–6, IEEE, 2017.

[60] J. G. Webster and H. Eren, *Measurement, instrumentation, and sensors handbook: spatial, mechanical, thermal, and radiation measurement*. CRC press, 2017.

[61] S. Mayya, P. Pierpaoli, G. Nair, and M. Egerstedt, "Collisions as information sources in densely packed multi-robot systems under mean-field approximations," in *Proc. Robot., Sci. Syst. Conf.*, 2017.

[62] U. Borrmann, L. Wang, A. D. Ames, and M. B. Egerstedt, "Control barrier certificates for safe swarm behavior," Georgia Institute of Technology, 2015.

[63] T. Schmickl, C. Möslinger, and K. Crailsheim, "Collective perception in a robot swarm," in *International Workshop on Swarm Robotics*, pp. 144–157, Springer, 2006.

[64] C. Scholtyssek, M. Dacke, R. Kröger, and E. Baird, "Control of self-motion in dynamic fluids: fish do it differently from bees," *Biology letters*, vol. 10, no. 5, p. 20140279, 2014.

[65] I. Schiffner, T. Perez, and M. V. Srinivasan, "Strategies for pre-emptive mid-air collision avoidance in budgerigars," *PloS one*, vol. 11, no. 9, p. e0162435, 2016.

[66] A. Lakhtakia and R. J. Martín-Palma, *Engineered biomimicry*. Newnes, 2013.

[67] A. Ugajin, T. Kiya, T. Kunieda, M. Ono, T. Yoshida, and T. Kubo, "Detection of neural activity in the brains of japanese honeybee workers during the formation of a "hot defensive bee ball"," *PLoS One*, vol. 7, no. 3, p. e32902, 2012.

[68] N. J. Mlot, C. A. Tovey, and D. L. Hu, "Fire ants self-assemble into waterproof rafts to survive floods," *Proceedings of the National Academy of Sciences*, vol. 108, no. 19, pp. 7669–7673, 2011.

[69] J. Löber, F. Ziebert, and I. S. Aranson, "Collisions of deformable cells lead to collective migration," *Scientific reports*, vol. 5, p. 9172, 2015.

[70] D. Grossman, I. Aranson, and E. B. Jacob, "Emergence of agent swarm migration and vortex formation through inelastic collisions," *New Journal of Physics*, vol. 10, no. 2, p. 023036, 2008.

[71] T. Schmickl, R. Thenius, C. Moeslinger, G. Radspieler, S. Kernbach, M. Szymanski, and K. Crailsheim, "Get in touch: cooperative decision making based on robot-to-robot collisions," *Autonomous Agents and Multi-Agent Systems*, vol. 18, no. 1, pp. 133–155, 2009.

[72] V. Radhakrishnan, "Locomotion: Dealing with friction," *Proceedings of the National Academy of Sciences*, vol. 95, no. 10, pp. 5448–5455, 1998.

[73] A. J. Rulison, J. L. Watkins, and B. Zambrano, "Electrostatic containerless processing system," *Review of scientific instruments*, vol. 68, no. 7, pp. 2856–2863, 1997.

[74] Y. Cai and D. Rote, "A review of dynamic stability of repulsive-force maglev suspension systems," tech. rep., Argonne National Lab., IL (United States), 1998.

[75] P. C. Nordine, J. R. Weber, and J. G. Abadie, "Properties of high-temperature melts using levitation," *Pure and applied chemistry*, vol. 72, no. 11, pp. 2127–2136, 2000.

[76] D. Foresti, M. Nabavi, M. Klingauf, A. Ferrari, and D. Poulikakos, "Acoustophoretic contactless transport and handling of matter in air," *Proceedings of the National Academy of Sciences*, vol. 110, no. 31, pp. 12549–12554, 2013.

[77] G. Guccione, M. Hosseini, S. Adlong, M. Johnsson, J. Hope, B. Buchler, and P. K. Lam, "Scattering-free optical levitation of a cavity mirror," *Physical review letters*, vol. 111, no. 18, p. 183001, 2013.

[78] "Frictional coefficients." Available at `https://en.wikibooks.org/w/index.php?title=Physics_Study_Guide&oldid=3289127`.

[79] R. L. Caldwell, "A unique form of locomotion in a stomatopod - backward somersaulting," *Nature*, vol. 282, pp. 71–73, nov 1979.

[80] "A female mantis shrimp (odontodactylus scyllarus)." Available at `https://www.nsf.gov/news/mmg/media/images/O_scyllarus_femalered_h.jpg`.

[81] L. G. Leal, *Advanced transport phenomena: fluid mechanics and convective transport processes*, vol. 7. Cambridge University Press, 2007.

[82] "San ace 40 9hv product information." Available at `http://www.sanyodenki.com/news/newslist/20150309_SanAce_40_9hv.html`.

[83] C. Wang, "Noise source analysis for two identical small axial-flow fans in series under operating condition," *Applied Acoustics*, vol. 129, pp. 13–26, 2018.

[84] "Trace width calculator." Available at `https://www.4pcb.com/trace-width-calculator.html`.

[85] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3400–3407, IEEE, 2011.

[86] A. S. Huang, E. Olson, and D. C. Moore, "Lcm: Lightweight communications and marshalling," in *Intelligent robots and systems (IROS), 2010 IEEE/RSJ international conference on*, pp. 4057–4062, IEEE, 2010.

[87] D. Meeker, "Finite element method magnetics," *FEMM*, vol. 4, p. 32, 2010.

[88] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," in *Readings in speech recognition*, pp. 159–165, Elsevier, 1990.

[89] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognition*, vol. 44, no. 3, pp. 678–693, 2011.

[90] M. Morel, C. Achard, R. Kulpa, and S. Dubuisson, "Time-series averaging using constrained dynamic time warping with tolerance," *Pattern Recognition*, vol. 74, pp. 77–89, 2018.

[91] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1–18, 2000.

[92] M. Senanayake, I. Senthooran, J. C. Barca, H. Chung, J. Kamruzzaman, and M. Murshed, "Search and tracking algorithms for swarms of robots: A survey," *Robotics and Autonomous Systems*, vol. 75, pp. 422–434, 2016.

[93] G. Valentini, D. Brambilla, H. Hamann, and M. Dorigo, "Collective perception of environmental features in a robot swarm," in *International Conference on Swarm Intelligence*, pp. 65–76, Springer, 2016.

[94] Y. Khaluf, "Edge detection in static and dynamic environments using robot swarms," in *2017 IEEE 11th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, pp. 81–90, IEEE, 2017.

[95] G. Mermoud, L. Matthey, W. C. Evans, and A. Martinoli, "Aggregation-mediated collective perception and action in a group of miniature robots," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 2-Volume 2*, pp. 599–606, International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[96] D. Tarapore, A. L. Christensen, P. U. Lima, and J. Carneiro, "Abnormality detection in multiagent systems inspired by the adaptive immune system," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 23–30, International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[97] G. Theraulaz and E. Bonabeau, "A brief history of stigmergy," *Artificial life*, vol. 5, no. 2, pp. 97–116, 1999.

Appendix A

How to Manufacture the HoverBot System

In this appendix, I explain the manufacture of the HoverBot system. I explain the manufacturing details of the magnet-levitation table in Section A.1, of the tethered and untethered prototypes in Section A.2, and of HoverBot versions 1 and 2 in Section A.3.

A.1 The Magnet-Levitation Table

I designed the magnet-levitation table design with Autodesk Inventor. The table consists of three different parts — one bottom surface, one top surface, and four side walls — which are shown in Figure A.1. I exported the Autodesk design and processed it in VCarve Pro. VCarve Pro is a CNC-routing software that is used to control CNC-machines such as the Shopbot Buddy. Shopbot Buddy is a compact CNC-machine that is designed for industrial, commercial, and residential applications cutting in wood, plastics, and aluminium. I screwed a sheet of 12.7mm thick MDF wood onto the bleeding board of a Shopbot buddy. I purchased the MDF sheet in a local hardware store. There are six individual wood pieces that assemble together to construct the magnet-levitation table. I drilled air-holes into the top surface of the magnet-levitation table using a 1.6mm drill bit. The following qrcode contains a link to a manufacturing video that I made while operating the Shopbot.

Using Shopbot to fabricate magnet-levitation table

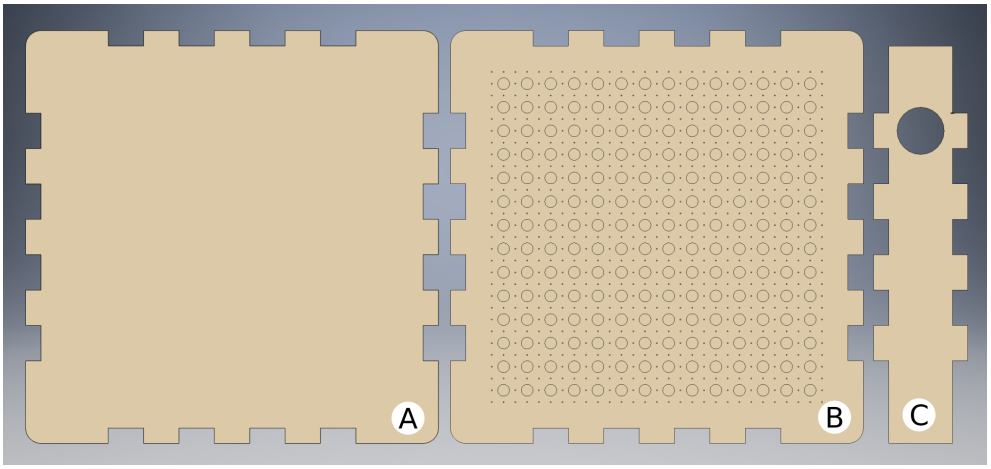


Figure A.1: The magnet-levitation table consists of three differently shaped layers: A) bottom surface; B) top surface; and C) side wall. The top surface has an array of air-holes and magnet pockets milled into its surface. The side wall has a fan inlet. All shapes have finger joints to increase the stability of the magnet-levitation table. Appendix C contains the technical drawings of the magnet-levitation table.

I drilled 12mm deep magnet pockets with Shopbot using a 10mm end mill. I finally used Shopbot to contour the top-, bottom-, and four side-walls out of the sheet of MDF. I put the technical drawings of the magnet-levitation table in Appendix C. The technical drawings contain all manufacturing relevant dimensions.

Before I assembled the table, I placed the top surface (see Figure A.1b) on a steel surface. I slid 0.063" wide and 0.394" thick neodymium (N42) magnets into the magnet pockets. I purchased the magnets from Amazon UK. I placed all magnets north-pole facing up. I explain this design choice in Section 3.2.3.4. The magnets are attracted to the steel surface; the interaction between magnet and steel surface overrules the interactions between adjacent magnets. Once the magnets were placed into the magnet pockets, I glued them into place with an Arrow TR400 hot glue gun. Before I glued the magnets, I verified their orientation by sliding a magnet with opposing polarity along the embedded magnets. All magnets should be assembled mono-directional (north-pole facing up).

Once the top surface was completed, I attached the four side walls to the bottom layer of the magnet-levitation table. I glued the walls into place using Gorilla Wood Glue which I purchased from a local hardware store. I flipped the construction upside down, so that the walls were supporting the bottom layer. I drilled three holes into the bottom layer and applied three T-Nut Leg Leveler which I purchased from Amazon USA. Leg leveler can be used for height adjustment. In Section 3.2.4.4, I discuss the importance of a completely levelled robot arena. I flipped the construction back and applied Gorilla Wood Glue to the top edges of the four side walls and inserted the top layer (smooth surface facing upwards); the finger joints helped increase the stability of the entire table. Once the glue dried, I mounted four San Ace 9HV0412P3K001 fans into the fan outlets using four standard wood screws per fan. I purchased the fans from Mouser.

Table A.1 contains the bill of materials. The materials for the magnet-levitation table cost $144 excluding manufacturing costs such as using Shopbot for manufacture.

Component	**Supplier**	**Cost($)**
Medium-density fibreboard wood	Hardware store	$20
N42 Disc Neodymium Magnets (14*14)	Amazon UK	$22
Gorilla wood glue	Hardware store	$10
Furniture levelers with threaded shank (20pcs)	Amazon USA	$12
San Ace 9HV0412P3K001 (x4)	Mouser	$80
Total		**$144**

Table A.1: Bill of materials of the magnet-levitation table.

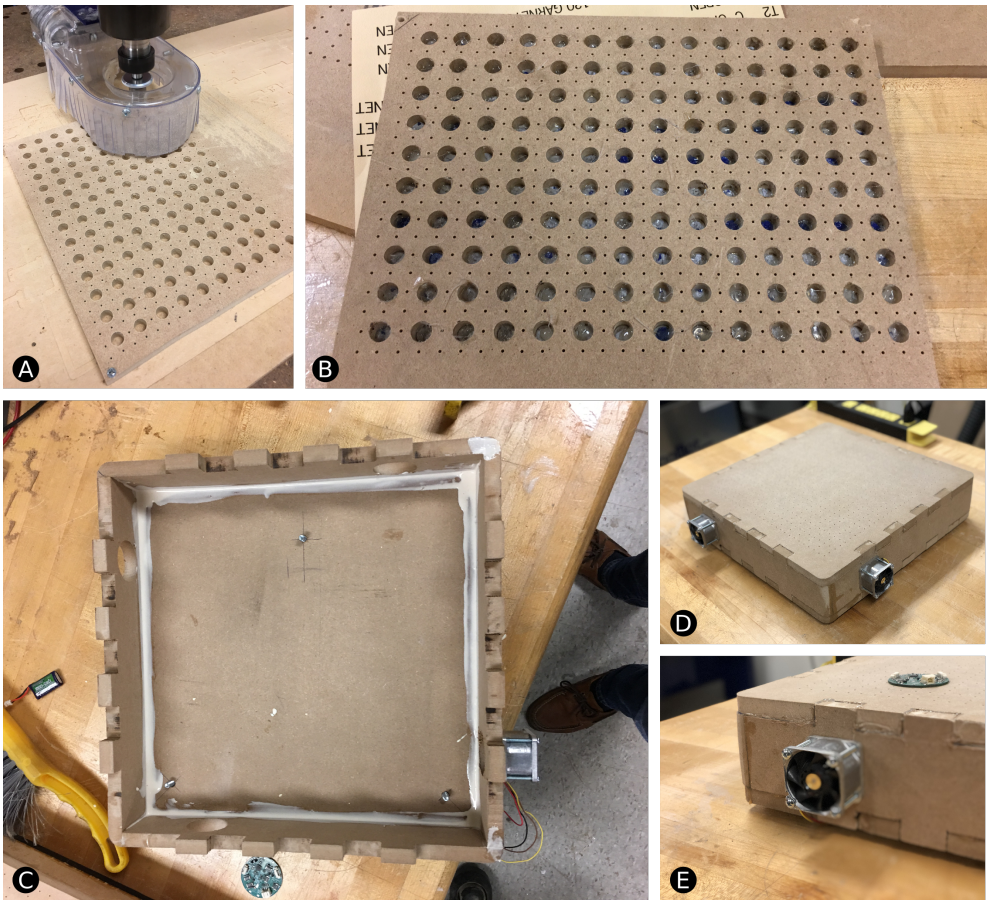


Figure A.2: Manufacturing collage of the magnet-levitation table. A) A CNC-machine processes a sheet of MDF. It drills air holes, mills magnet pockets, and contours all six table parts. B) Magnets are inserted into pockets and glued into place. C) The four side-walls and the bottom layer are glued first. D) The top surface follows. E) Four fans are attached; one fan per side-wall.

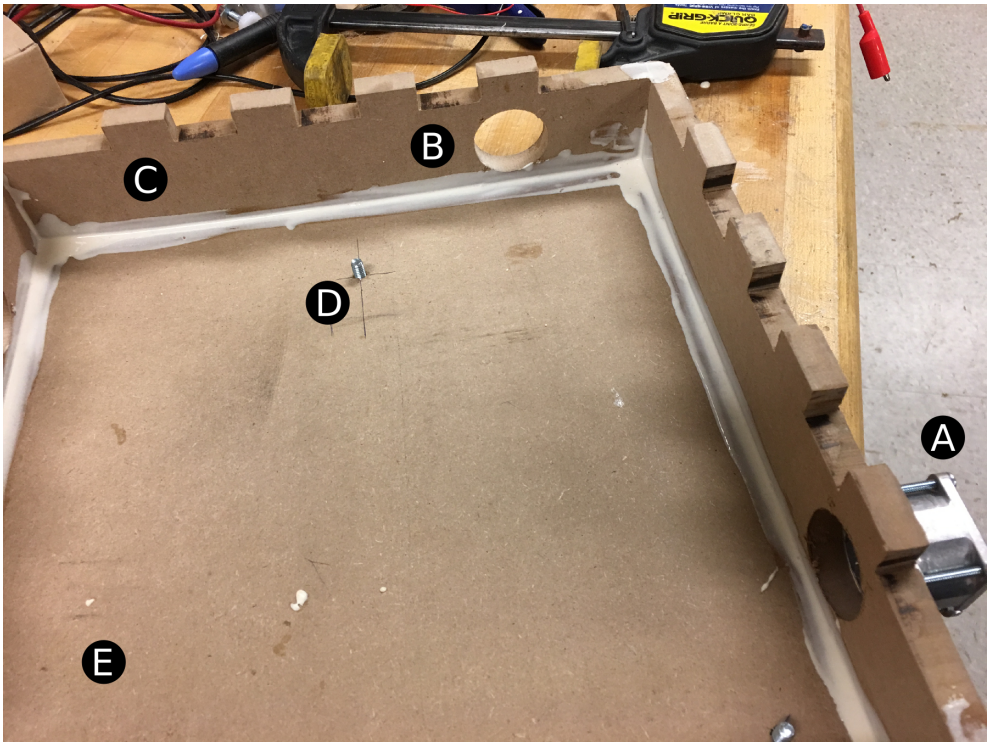


Figure A.3: The bottom surface and side walls are assembled and glued together. A) fan; B) fan inlet; C) side-wall; D) screw from leg leveler; and E) bottom surface.

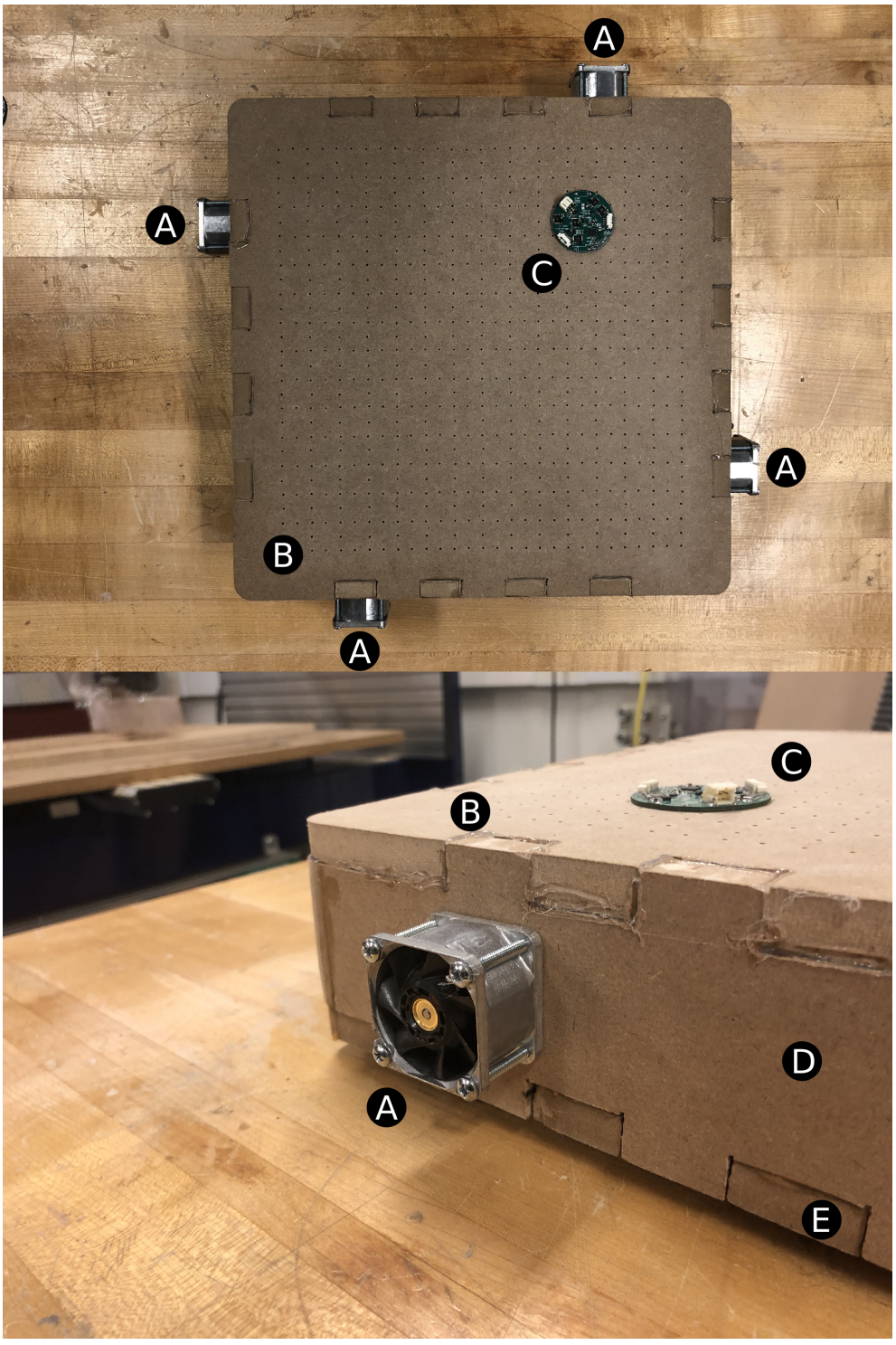


Figure A.4: Top and side view of the magnet-levitation table. A) Fan; B) top surface; C) HoverBot; D) side-wall; and E) bottom surface.

A.2 Tethered and Untethered Prototypes

Figure A.5 shows the schematic and board design of the untethered prototype. The tethered prototype only consists of the coil layer shown in Figure A.5b. The PCB material is a FR4-Cu composite with $1\frac{\text{oz.}}{\text{ft}^2}$. I used a Protolaser U3 micro-machining system to manufacture $50\mu\text{m}$ coil traces. I used a Weller soldering iron to solder the electronics components. The electronics components consisted of an *ATtiny84* microcontroller, an *ADP123* adjustable voltage regulator, three *FDC6401N* Dual N-channel MOSFETs, and Turnigy Nano-tech 300mAh lithium polymer batteries. I used 0.224mm insulated copper wire to connect coils with coil pin headers. I used a lighter to take off the insulation layer from the copper wire endings and used sandpaper to increase their contact roughness before soldering them to the coils and connecting them to the coil pin headers. I purchased the copper wire and electronics components from *RS Components* and the batteries from *Hobby King*.

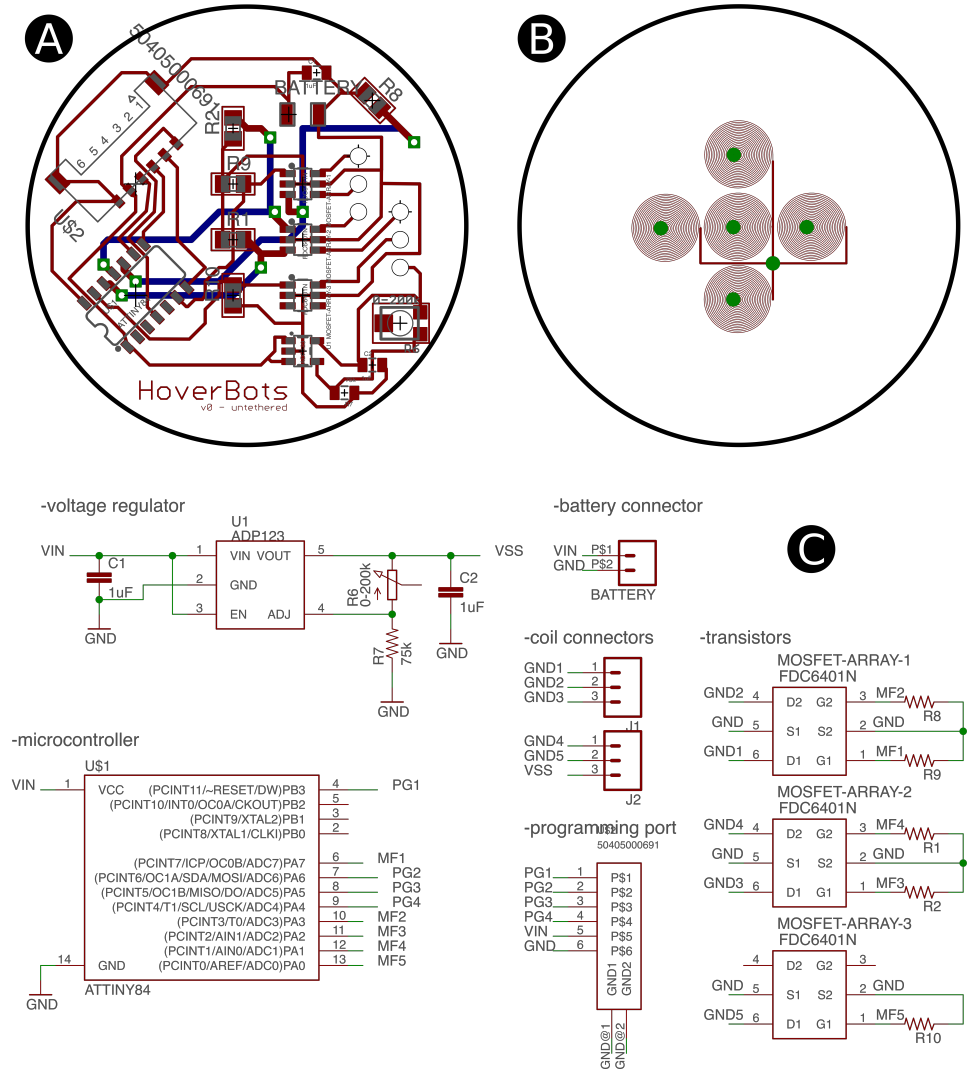


Figure A.5: Untethered prototype schematic and board design. A) top PCB; B) bottom PCB; and C) schematic of the top PCB.

A.3 HoverBot Versions 1 and 2

Figure A.6 shows the Eagle library of HoverBot's planar coils. I designed HoverBots with Autodesk Eagle. They meet the minimum specifications of *Advanced Circuits*, an American PCB manufacturer. Figure A.7 shows the PCB specifications of *Advanced Circuits* for barebone, 2-layer, and 4-layer PCBs. There is a student option that lowers the minimum order quantity to a single board. The planar coils are 13mm in diameter and are embedded into the bottom layer of the PCB. Each coil possesses 17 turns, a trace width of $150\,\mu\text{m}$, and a trace thickness of $1\,\frac{\text{oz}}{\text{ft}^2}$. I originally designed 13mm planar coils to match the permanent magnets that I planned on purchasing. However, I ended up buying 10mm diameter magnets, and keeping the 13mm coils. I discuss in Section 3.2.3.2 the relationship between coil diameter and magnet size. HoverBots version 1 and 2 are 43mm in diameter.

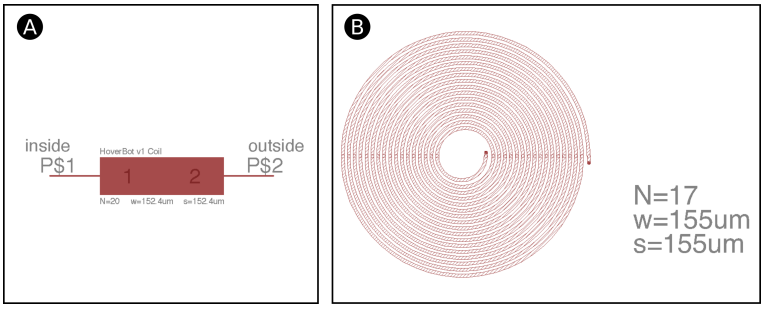


Figure A.6: Eagle library of HoverBot's planar coil. I designed a planar coil library for HoverBots. I used this planar coil design for HoverBot versions 1 and 2.

The manufacture of HoverBot versions 1 and 2 is identical. They only differ in functionality. HoverBot consists of a four layer PCB — three ground layers and one 3.3V layer — which are shown in Figure A.8. The ground and voltage layers help route the components. If a component has to be connected to ground or supply voltage, I insert a via (vertical interconnect access) close to the component's pin. Autodesk Eagle automatically connects the component pin with the corresponding layer. I put additional ground vias all over the board to ensure efficient current paths back to ground. The only traces I could not fit onto the top PCB layer were the traces between JTAG connector and microcontroller and shutdown signal of the H-bridges and microcontroller (version 2). I routed those signals on the 3.3V PCB layer. Some components

BareBones®	**$33**Each	**$66**Each
2 Layer - **1 Day** Turn	2 Layer - **5 Day** Turn	4 Layer - **5 Day** Turn
10" x 16" Max Board Size	**Max. Board Size: *60* sq. in.**	**Max. Board Size: 30 sq. in.**
Min. Order Quantity: 1	Min. Order Quantity: 3	Min. Order Quantity: 4
FR-4 .062" Material	FR-4 .062" Material	FR-4 .062" Material
I oz. Cu.	I oz. Cu.	I oz. Cu.
Tin Finish	Lead-Free Solder Finish*	Lead-Free Solder Finish*
No Mask *(bare)*	Green Mask	Green Mask
No Legend	White Legend (1 or 2 Sides)	White Legend (1 or 2 Sides)
Custom Shape[1]	Custom Shape	Custom Shape

Figure A.7: Advanced Circuit pricing options. Advanced Circuit is an American PCB foundry.

possess uncommon or even unique footprints. In such cases, I studied the component's datasheet and manually designed the footprint. The smaller the component's footprint, the more difficult it is to solder. The smallest components that I used in my designs are *0402* components. *0402* refers to a standard surface-mount package; components of that package are 0.4mm long and 0.2mm wide.

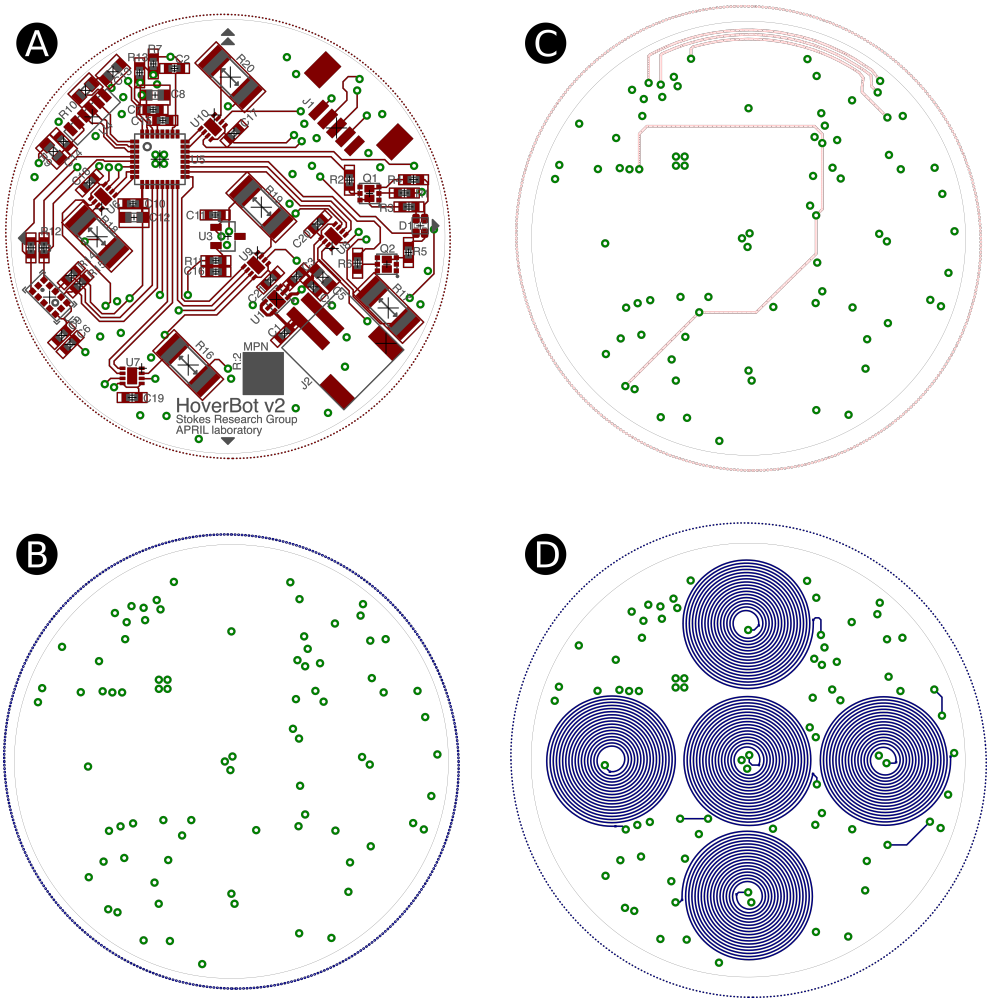


Figure A.8: The four PCB layers of HoverBot versions 1 and 2. A) The top layer is connected to ground and mounts all of HoverBot's components. B) The second layer is connected to ground. C) The third layer is connected to 3.3V. D) The fourth layer is connected to ground and incorporates the planar coil array.

I ordered all Hover-designs as *panelised* 4-layer PCBs. A panelised PCB is a sheet that contains several copies of the same design. You have to manually cut out the individual boards. I used a bandsaw to cut out the boards and a dual wheeled bench grinder to take off additional material until the boards were circular discs. Figure A.9 exemplarily shows a panelised board full of HoverTags and individual cut-outs. It is usually cheaper to panelise a design than to order individual boards. Figure A.11 shows the Eagle board design of HoverBot version 2. Figure A.10 shows the schematics of HoverBot version 2. I put the bill of materials for HoverBot version 2 in Appendix B. HoverBot version 1 is mostly a subset of HoverBot version 2.

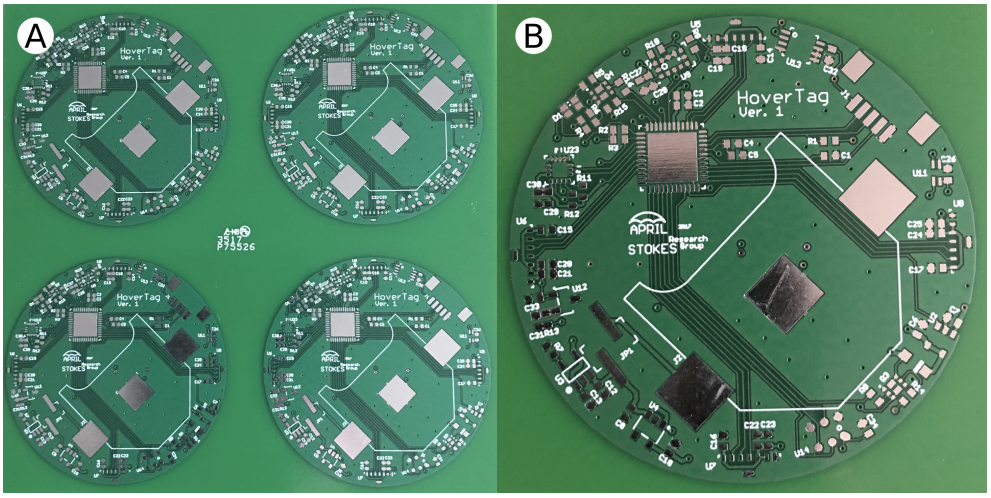


Figure A.9: Panelised PCB design. A) A single PCB design contains several copies of the same circuit board. B) The boards have to be cut into individual quadrants. I used a bandsaw to cut them out and a dual wheeled bench grinder to take off additional material until they became circular discs.

The following qrcode contains a link to a video that shows the manual assembly of HoverBot version 2. It shows the entire process from populating the PCB to programming it.

Video: Manufacturing HoverBots

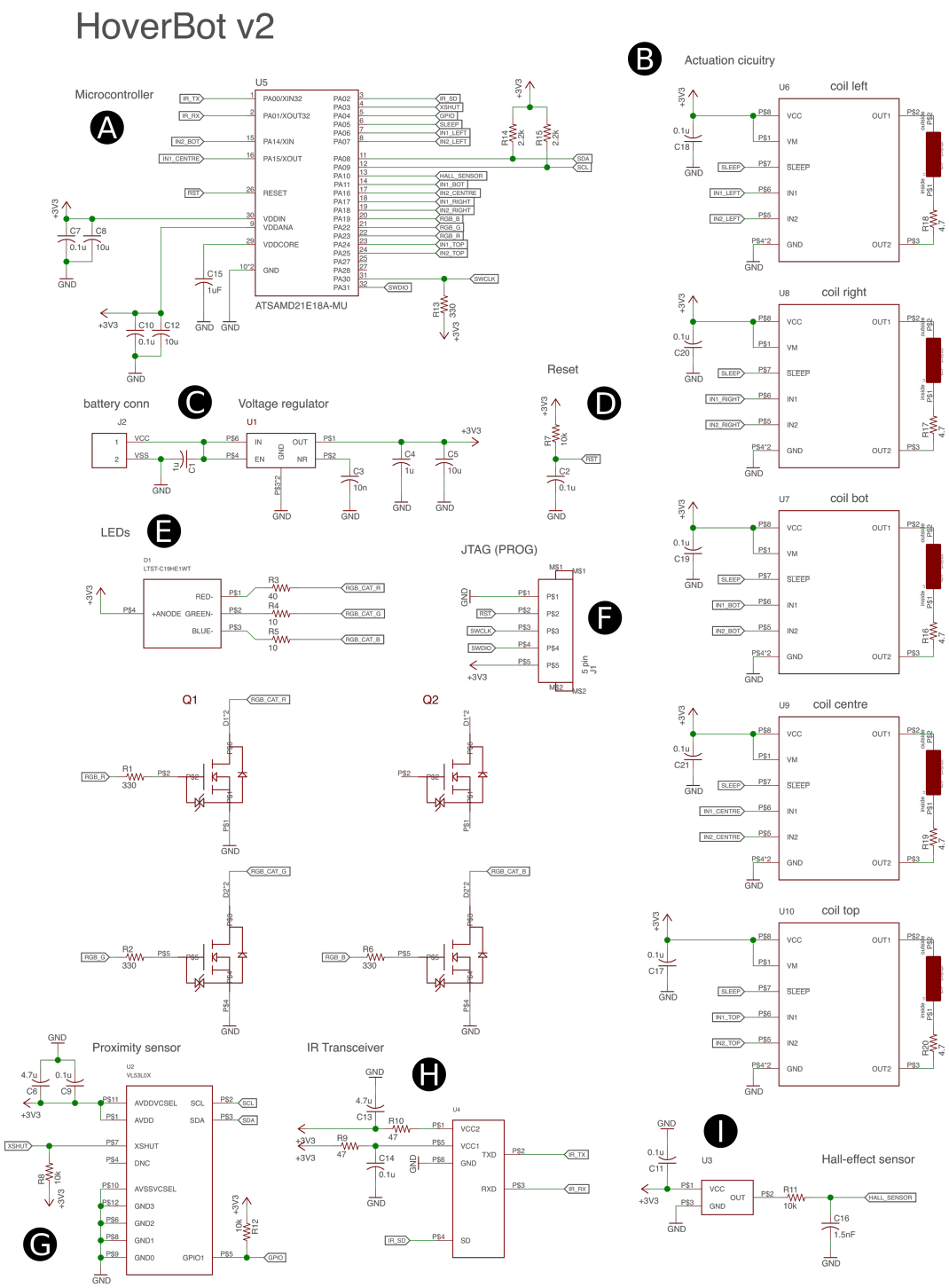


Figure A.10: Circuit schematic of HoverBot version 2. A) microcontroller; B) H-bridge drivers; C) voltage regulator; D) reset circuit; E) RGB LED; F) JTAG programming port; G) proximity/distance sensor pointing upwards; H) infrared transceiver; and I) Hall-effect sensor.

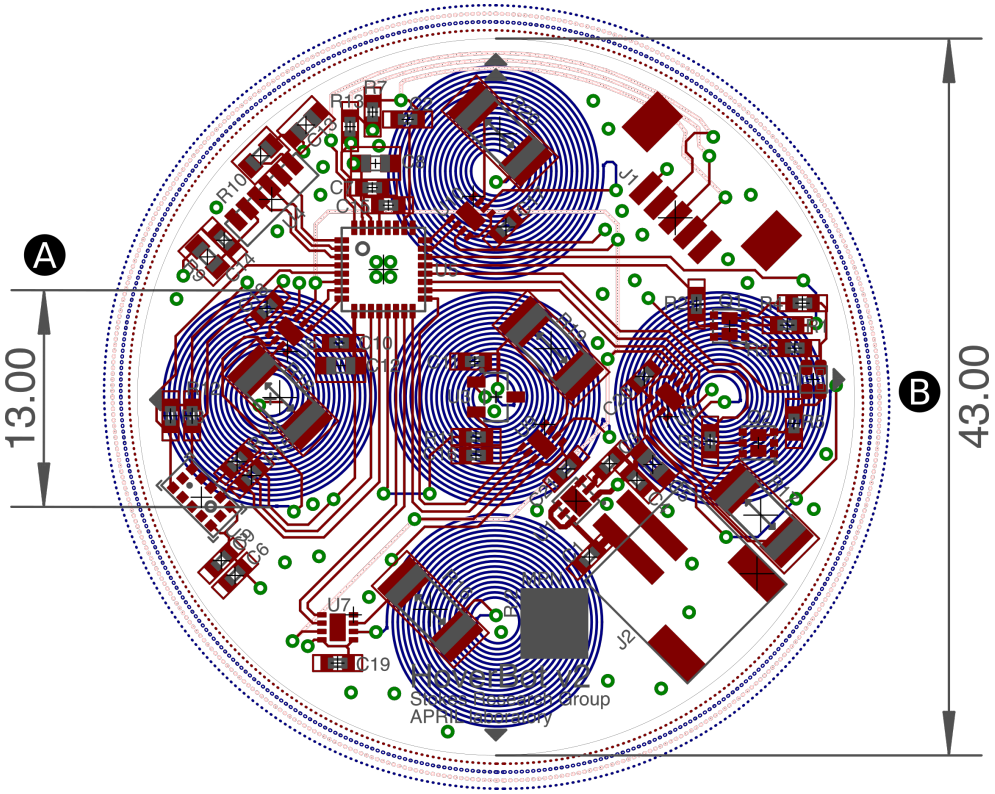


Figure A.11: Board design of HoverBot version 2.

Appendix B

Bill of Materials for HoverBot Version 2

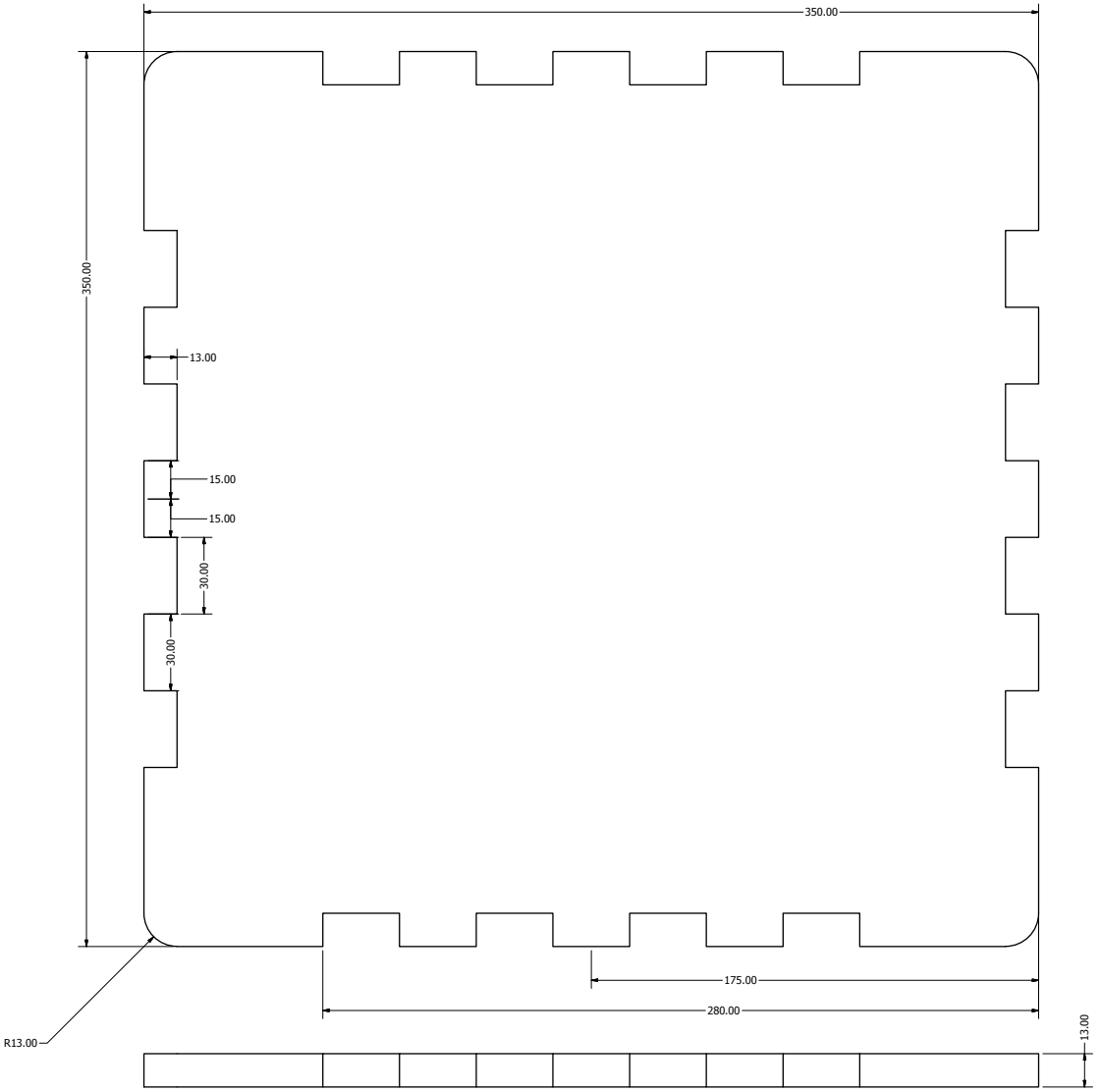
Title: Bill of materials for HoverBot version 2 in quantities of 1s and 10s
Designer: Markus P. Nemitz

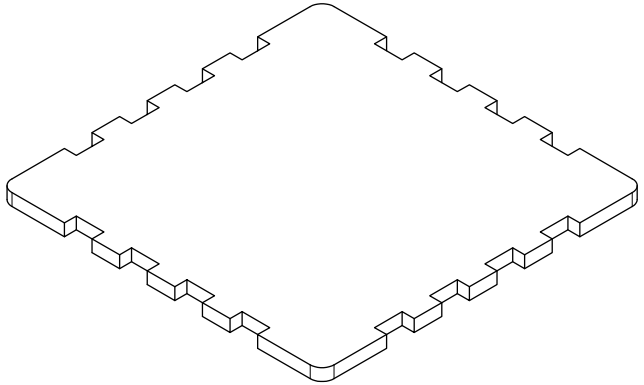
Reference	# for function	# per board	Value	Distributer	Part number	Footprint	Cost @1	Total Cost	Cost @10	Total Cost
Computation										
U5; main MCU	1	1	ATSAMD21E16B-MU	Mouser	556-ATSAMD21E16B-MU	48-QFN	**$ 1.610**	$ 1.61	**$ 1.610**	$ 1.61
C7, C10; bypass capacitors	2	6	0.1uF	Digi-key	445-7344-1-ND	,0402	**$ 0.110**	$ 0.66	**$ 0.078**	$ 0.47
C8, C12; bypass capacitors	2	6	10uF	Digi-key	490-10693-1-ND	,0402	**$ 0.130**	$ 0.78	**$ 0.094**	$ 0.56
C15; bypass capacitor	1	3	1uF	Digi-key	587-1454-1-ND	,0402	**$ 0.120**	$ 0.36	**$ 0.090**	$ 0.27
R13; SWCLK, pull-up resistor	1	3	330 ohm, 0.1W	Digi-key	P330JCT-ND	,0402	$ 0.100	$ 0.30	$ 0.026	$ 0.08
R14; SDA, pull-up resistor	1	3	2.2 kohm, 0.1W	Digi-key	P2.2KJCT-ND	,0402	**$ 0.100**	$ 0.30	**$ 0.030**	$ 0.09
R15; SCL, pull-up resistor	1	3	2.2 kohm, 0.1W	Digi-key	P2.2KJCT-ND	,0402	**$ 0.100**	$ 0.30	**$ 0.030**	$ 0.09
Power										
Battery	1	1	300mAh LiPo	HobbyKing	9210000168-0	34x20x7.5mm	**$ 3.440**	$ 3.44	**$ 3.440**	$ 3.44
U1; Voltage Regulator	1	1	3V3	Digi-key	296-39488-1-ND	6-WDFN	**$ 1.780**	$ 1.78	**$ 1.602**	$ 1.60
C3; bypass capacitors	1	3	10n	Digi-key	399-3066-1-ND	,0402	**$ 0.100**	$ 0.30	**$ 0.031**	$ 0.09
C1, C4; bypass capacitors	2	6	1uF	Digi-key	587-1454-1-ND	,0402	**$ 0.120**	$ 0.72	**$ 0.090**	$ 0.54
C5; bypass capacitors	1	3	10uF	Digi-key	490-10693-1-ND	,0402	**$ 0.130**	$ 0.39	**$ 0.094**	$ 0.28
Communication										
U4; IR transceiver	1	1	SERCOM	Digi-key	751-1066-1-ND	custom	**$ 4.090**	$ 4.09	**$ 3.677**	$ 3.68
R9, R10; IrDA current-limiting resistors	2	6	47 ohm, 0.125W	Digi-key	MCT0603-47.0-CFCT-ND	,0603	$ 0.270	$ 1.62	$ 0.229	$ 1.37
C13; filter capacitor	1	3	4.7uF	Digi-key	490-10481-1-ND	,0603	$ 0.100	$ 0.30	$ 0.013	$ 0.04
C14; bypass capacitor	1	3	0.1uF	Digi-key	445-7344-1-ND	,0402	$ 0.100	$ 0.30	$ 0.013	$ 0.04
Sensing										
U2; Proximity	1	1	I2C	Digi-key	497-16538-1-ND	custom	$ 5.460	$ 5.46	$ 4.903	$ 4.90
C6; bypass capacitor	1	3	4.7uF	Digi-key	1276-1483-1-ND	,0402	$ 0.140	$ 0.42	$ 0.097	$ 0.29
C9; bypass capacitor	1	3	0.1uF	Digi-key	445-7344-1-ND	,0402	$ 0.100	$ 0.30	$ 0.013	$ 0.04
R8, R12; pull-up resistors	2	6	10k	Digi-key	P10KJCT-ND	,0402	$ 0.100	$ 0.60	$ 0.029	$ 0.17
U3; Magnetic Sensor	1	1	ADC	Digi-key	296-38457-1-ND	custom	$ 1.250	$ 1.25	$ 0.976	$ 0.98
C11; bypass capacitor	1	3	0.1uF	Digi-key	445-7344-1-ND	,0402	**$ 0.110**	$ 0.33	**$ 0.078**	$ 0.23
C16; low-pass capacitor	1	3	1.5nF	Digi-key	490-3245-1-ND	,0402	**$ 0.100**	$ 0.30	**$ 0.013**	$ 0.04
R11; low-pass resistor	1	3	10k	Digi-key	P10KJCT-ND	,0402	$ 0.100	$ 0.30	$ 0.029	$ 0.09
Locomotion										
U6, U7, U8, U9, U10; NMOS Half-bridge	5	5	Half-bridge	Digi-key	296-47205-1-ND	custom	$ 1.050	$ 5.25	$ 0.939	$ 4.70
C17, C18, C19, C20, C21; bypass capacitors	5	15	0.1uF	Digi-key	445-7344-1-ND	,0402	**$ 0.110**	$ 1.65	**$ 0.078**	$ 1.17
R16, R17, R18, R19, R20; CLRs resitors	5	25	4.7 ohm, 1W	Digi-key	R16, R17, R18, R19, R20	,2010	**$ 0.750**	$ 18.75	**$ 0.555**	$ 13.88
User I/O										
D1; RGB LED	1	1	PWM	Digi-key	160-2162-1-ND	0606(1616 M)	**$ 0.590**	$ 0.59	**$ 0.424**	$ 0.42
Q1, Q2; NMOS Switching Pair	2	2	Dual NMOS Pair	Digi-key	SSM6N58NULFCT-ND	custom	**$ 0.560**	$ 1.12	**$ 0.439**	$ 0.88
R1, R2, R6; CLRs resistors	3	9	330 ohm, 0.1W	Digi-key	P330JCT-ND	,0402	$ 0.100	$ 0.30	$ 0.026	$ 0.08
R3; CLRs resistors	1	3	43 ohm, 0.1W	Digi-key	43JCT-ND	,0402	$ 0.100	$ 0.30	$ 0.029	$ 0.09
R4, R5; CLRs resistors	2	6	10 ohm, 0.1W	Digi-key	P10.7LCT-ND	,0402	$ 0.100	$ 0.60	$ 0.023	$ 0.14
Connectors										
J1; Molex Connector	1	1	JTAG	Digi-key	WM7623CT-ND	custom	**$ 1.360**	$ 1.36	**$ 1.200**	$ 1.20
J2; Battery Connector	1	1		Digi-key	455-1749-1-ND	custom	**$ 0.580**	$ 0.58	**$ 0.547**	$ 0.55
R7; Reset	1	3	10k	Digi-key	P10KJCT-ND	,0402	$ 0.100	$ 0.30	$ 0.029	$ 0.09
C2; Reset	1	3	0.1uF	Digi-key	445-7344-1-ND	,0402	$ 0.100	$ 0.30	$ 0.013	$ 0.04
							Total Total	$ 57.31	Total Total	$ 44.22

Appendix C

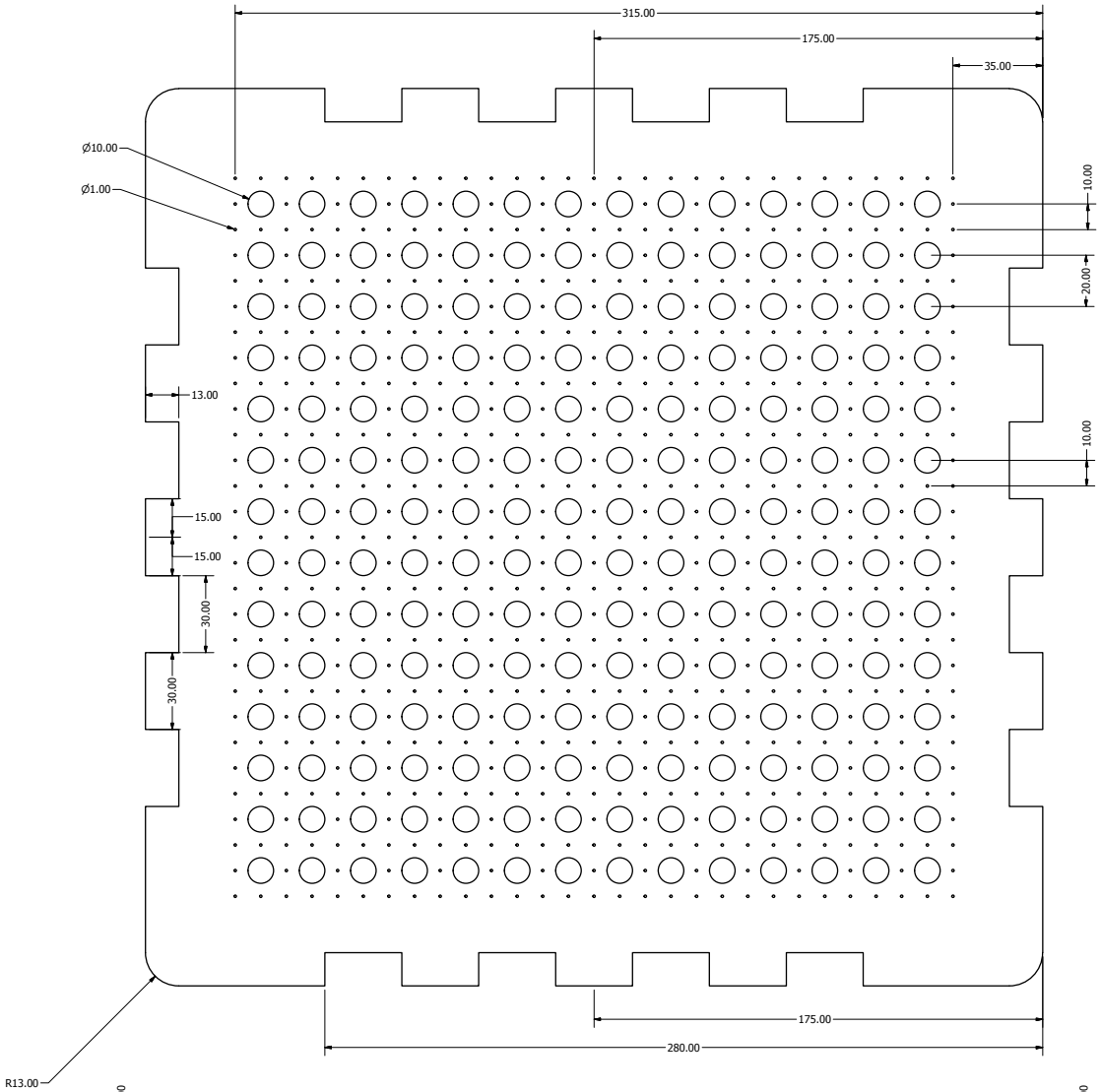
Technical Drawings of Magnet-Levitation Table

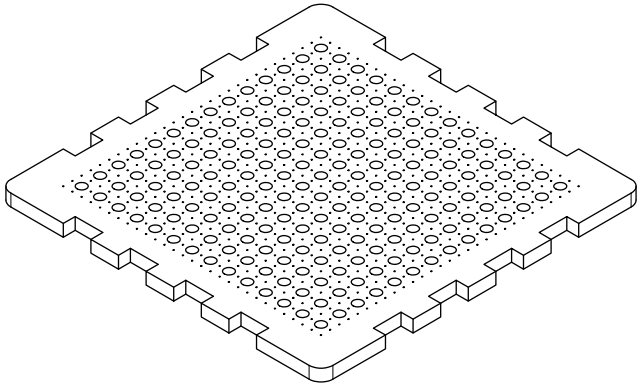
Title: Technical drawings of the magnet-levitation table
Components: Bottom surface, top surface, and side walls
Designer: Markus P. Nemitz



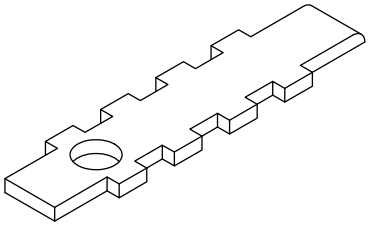


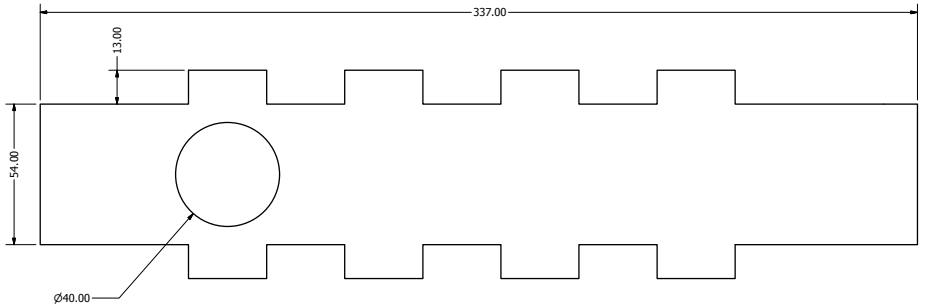
DRAWN Markus P. Nemitz	09/08/2018	The University of Edinburgh, Stokes Research Group
CHECKED		TITLE: Magnet-levitation table - bottom surface
QA		
MFG		
APPROVED		
		SIZE D · DWG NO bottom_surface-1 · REV
		SCALE 1 : 1 · SHEET 1 OF 1

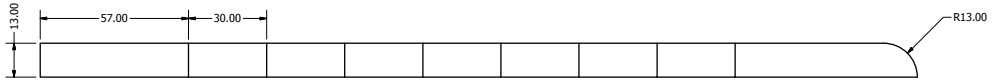




DRAWN Markus P. Nemitz	09/08/2018	The University of Edinburgh, Stokes Research Group
CHECKED		TITLE: Magnet-levitation table - top surface
QA		
MFG		
APPROVED		
	SIZE D	DWG NO top_surface-2 REV
	SCALE 1 : 1	SHEET 1 OF 1







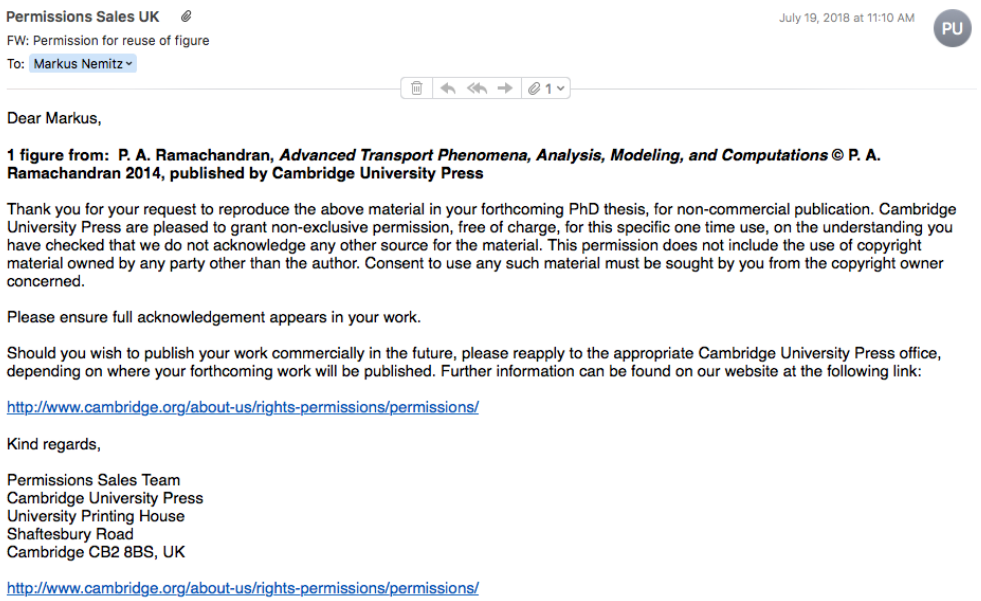
DRAWN Markus P. Nemitz	09/08/2018	The University of Edinburgh, Stokes Research Group
CHECKED		TITLE: Magnet-levitation table - side wall
QA		
MFG		
APPROVED		
SIZE D	DWG NO side_wall_hole	REV
SCALE 1 : 1	SHEET 1 OF 1	

Appendix D

Copyright Agreements

D.1 Cambridge University Press

Book: Advanced Transport Phenomena, Analysis, Modeling, and Computations
Author: P. A. Ramachandran (2014)
Publisher: Cambridge University Press



Permissions Sales UK July 19, 2018 at 11:10 AM

FW: Permission for reuse of figure

To: Markus Nemitz

Dear Markus,

1 figure from: P. A. Ramachandran, *Advanced Transport Phenomena, Analysis, Modeling, and Computations* © P. A. Ramachandran 2014, published by Cambridge University Press

Thank you for your request to reproduce the above material in your forthcoming PhD thesis, for non-commercial publication. Cambridge University Press are pleased to grant non-exclusive permission, free of charge, for this specific one time use, on the understanding you have checked that we do not acknowledge any other source for the material. This permission does not include the use of copyright material owned by any party other than the author. Consent to use any such material must be sought by you from the copyright owner concerned.

Please ensure full acknowledgement appears in your work.

Should you wish to publish your work commercially in the future, please reapply to the appropriate Cambridge University Press office, depending on where your forthcoming work will be published. Further information can be found on our website at the following link:

http://www.cambridge.org/about-us/rights-permissions/permissions/

Kind regards,

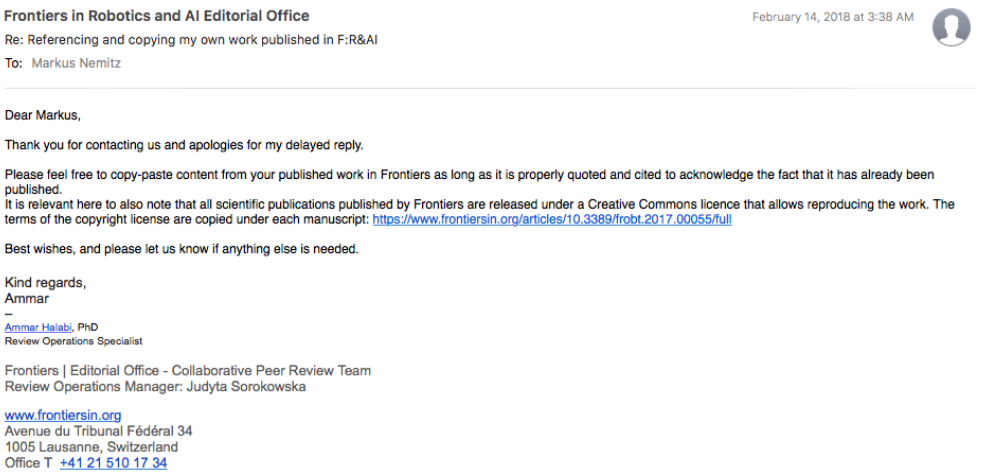
Permissions Sales Team
Cambridge University Press
University Printing House
Shaftesbury Road
Cambridge CB2 8BS, UK

http://www.cambridge.org/about-us/rights-permissions/permissions/

Figure D.1: Permission to reuse figure. Cambridge University Press.

D.2 Frontiers in Robotics and Artificial Intelligence

Journal: Frontiers in Robotics and Artificial Intelligence
Author: Nemitz et al. (2017, 2018)



Frontiers in Robotics and AI Editorial Office February 14, 2018 at 3:38 AM

Re: Referencing and copying my own work published in F:R&AI

To: Markus Nemitz

Dear Markus,

Thank you for contacting us and apologies for my delayed reply.

Please feel free to copy-paste content from your published work in Frontiers as long as it is properly quoted and cited to acknowledge the fact that it has already been published.
It is relevant here to also note that all scientific publications published by Frontiers are released under a Creative Commons licence that allows reproducing the work. The terms of the copyright license are copied under each manuscript: https://www.frontiersin.org/articles/10.3389/frobt.2017.00055/full

Best wishes, and please let us know if anything else is needed.

Kind regards,
Ammar
–
Ammar Halabi, PhD
Review Operations Specialist

Frontiers | Editorial Office - Collaborative Peer Review Team
Review Operations Manager: Judyta Sorokowska

www.frontiersin.org
Avenue du Tribunal Fédéral 34
1005 Lausanne, Switzerland
Office T +41 21 510 17 34

Figure D.2: Permission to reuse content of my Frontiers publications. Frontiers in Robotics and Artificial Intelligence.

D.3 Springer Nature

Book: Swarm Robotics: A Formal Approach
Author: Heiko Hamann (2018)
Publisher: Springer Nature

SPRINGER NATURE LICENSE TERMS AND CONDITIONS

Aug 05, 2018

This Agreement between The University of Edinburgh -- Markus Nemitz ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	4402660023763
License date	Aug 05, 2018
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	Introduction to Swarm Robotics
Licensed Content Author	Heiko Hamann
Licensed Content Date	Jan 1, 2018
Type of Use	Thesis/Dissertation
Requestor type	non-commercial (non-profit)
Format	print and electronic
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	1
Will you be translating?	no
Circulation/distribution	<501
Author of this Springer Nature content	no
Title	HoverBot: A Manufacturable Swarm Robot that Has Multi-functional Sensing Capabilities, and which Uses Collisions for Two-dimensional Mapping
Instructor name	Dr. Adam A. Stokes
Institution name	The University of Edinburgh
Expected presentation date	Sep 2018
Portions	Figure 1.1
Requestor Location	The University of Edinburgh Alexander Crum Brown Rd Edinburgh, EH9 3JQ United Kingdom Attn: The University of Edinburgh
Billing Type	Invoice
Billing Address	The University of Edinburgh Alexander Crum Brown Rd Edinburgh, United Kingdom EH9 3JQ Attn: The University of Edinburgh
Total	0.00 USD
Terms and Conditions	

Springer Nature Terms and Conditions for RightsLink Permissions

Springer Customer Service Centre GmbH (the Licensor) hereby grants you a non-exclusive, world-wide licence to reproduce the material and for the purpose and requirements specified in the attached copy of your order form, and for no other use, subject to the conditions below:

1. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of this material. However, you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

   If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also seek permission from that source to reuse the material.

2. Where **print only** permission has been granted for a fee, separate permission must be obtained for any additional electronic re-use.

3. Permission granted **free of charge** for material in print is also usually granted for any electronic version of that work, provided that the material is incidental to your work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.

4. A licence for 'post on a website' is valid for 12 months from the licence date. This licence does not cover use of full text articles on websites.

5. Where **'reuse in a dissertation/thesis'** has been selected the following terms apply:

Print rights for up to 100 copies, electronic rights for use only on a personal website or institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/).

6. Permission granted for books and journals is granted for the lifetime of the first edition and does not apply to second and subsequent editions (except where the first edition permission was granted free of charge or for signatories to the STM Permissions Guidelines http://www.stm-assoc.org/copyright-legal-affairs/permissions/permissions-guidelines/), and does not apply for editions in other languages unless additional translation rights have been granted separately in the licence.

7. Rights for additional components such as custom editions and derivatives require additional permission and may be subject to an additional fee. Please apply to Journalpermissions@springernature.com/bookpermissions@springernature.com for these rights.

8. The Licensor's permission must be acknowledged next to the licensed material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract, and must be hyperlinked to the journal/book's homepage. Our required acknowledgement format is in the Appendix below.

9. Use of the material for incidental promotional use, minor editing privileges (this does not include cropping, adapting, omitting material or any other changes that affect the meaning, intention or moral rights of the author) and copies for the disabled are permitted under this licence.

10. Minor adaptations of single figures (changes of format, colour and style) do not require the Licensor's approval. However, the adaptation should be credited as shown in Appendix below.

Appendix — Acknowledgements:

For Journal Content:
Reprinted by permission from [**the Licensor**]: [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication)

For **Advance Online Publication papers:**
Reprinted by permission from [**the Licensor**]: [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM].)

For Adaptations/Translations:
Adapted/Translated by permission from [**the Licensor**]: [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication)

Note: For any republication from the British Journal of Cancer, the following credit line style applies:

Reprinted/adapted/translated by permission from [**the Licensor**]: on behalf of Cancer Research UK: : [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication)

For **Advance Online Publication** papers:
Reprinted by permission from The [**the Licensor**]: on behalf of Cancer Research UK: [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM])

For Book content:
Reprinted/adapted by permission from [**the Licensor**]: [**Book Publisher** (e.g. Palgrave Macmillan, Springer etc) [**Book Title**] by [**Book author**(s)] [**COPYRIGHT**] (year of publication)

Other Conditions:

Version 1.0

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Appendix E

Journal: HoverBots: Precise Locomotion Using Robots That Are Designed for Manufacturability

Journal: Frontiers in Robotics and Artificial Intelligence

Section: Multi-Robot Systems

Research topic: Novel Technological and Methodological Tools for the Understanding of Collective Behaviours

Prof. Simon Garnier *editor*
Assistant Professor in Robotics
Department of Biological Sciences
New Jersey Institute of Technology, USA

Dr. Sabine Hauert *reviewer*
Lecturer in Robotics
Engineering Mathematics
University of Bristol, United Kingdom

Dr.-Ing. Heiko Hamann *reviewer*
Professor for Service Robotics
Institute of Computer Engineering
University of Lubeck, Germany

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 06 November 2017
doi: 10.3389/frobt.2017.00055

HoverBots: Precise Locomotion Using Robots That Are Designed for Manufacturability

Markus P. Nemitz[1,2], Mohammed E. Sayed[1], John Mamish[2], Gonzalo Ferrer[2], Lijun Teng[1], Ross M. McKenzie[1], Alfred O. Hero[2], Edwin Olson[2] and Adam A. Stokes[1]**

[1] School of Engineering, Institute for Integrated Micro and Nano Systems, The University of Edinburgh, Edinburgh, United Kingdom, [2] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, United States

Scaling up robot swarms to collectives of hundreds or even thousands without sacrificing sensing, processing, and locomotion capabilities is a challenging problem. Low-cost robots are potentially scalable, but the majority of existing systems have limited capabilities, and these limitations substantially constrain the type of experiments that could be performed by robotics researchers. As an alternative to increasing the quantity of robots by reducing their functionality, we have developed a new technology that delivers increased functionality at low-cost. In this study, we present a comprehensive literature review on the most commonly used locomotion strategies of swarm robotic systems. We introduce a new type of low-friction locomotion—active low-friction locomotion—and we show its first implementation in the HoverBot system. The HoverBot system consists of an air levitation and magnet table, and a HoverBot agent. HoverBot agents are levitating circuit boards that we have equipped with an array of planar coils and a Hall-effect sensor. The HoverBot agent uses its coils to pull itself toward magnetic anchors that are embedded into a levitation table. These robots use active low-friction locomotion; consist of only surface-mount components; circumvent actuator calibration; are capable of odometry by using a single Hall-effect sensor; and perform precise movement. We conducted three hours of experimental evaluation of the HoverBot system in which we observed the system performing more than 10,000 steps. We also demonstrate formation movement, random collision, and straight collisions with two robots. This study demonstrates that active low-friction locomotion is an alternative to wheeled and slip-stick locomotion in the field of swarm robotics.

Keywords: HoverBot, swarm robots, design for manufacturability, low-friction locomotion, precise locomotion, robot testbed, physical simulation

OPEN ACCESS

Edited by:
Simon Garnier, New Jersey Institute of Technology, United States

Reviewed by:
Sabine Hauert, University of Bristol, United Kingdom
Heiko Hamann, University of Lübeck, Germany

****Correspondence:***
Markus P. Nemitz
m.nemitz@ed.ac.uk, nemitz@umich.edu;
Adam A. Stokes
a.a.stokes@ed.ac.uk

Specialty section:
This article was submitted to Multi-Robot Systems, a section of the journal Frontiers in Robotics and AI

Received: *08 June 2017*
Accepted: *13 October 2017*
Published: *06 November 2017*

Citation:
Nemitz MP, Sayed ME, Mamish J, Ferrer G, Teng L, McKenzie RM, Hero AO, Olson E and Stokes AA (2017) HoverBots: Precise Locomotion Using Robots That Are Designed for Manufacturability. Front. Robot. AI 4:55. doi: 10.3389/frobt.2017.00055

INTRODUCTION

Swarm robotics is the study of developing and controlling scalable groups of simple robots. Individual robots within a swarm only possess limited capabilities. They move in two- or three-dimensional space, sense their local environment, and communicate with only their nearest neighbors. These local interactions between hundreds or thousands of robots can potentially give rise to complex behaviors (Brambilla et al., 2013). Much swarm robotics research is inspired by the observation of emergent behaviors in nature (Bonabeau et al., 1999). Colonies of termites work together to build complex structures that are of great importance for survival of the colony as a whole. Schools of fish cluster together making it difficult for a visually orientated predator to pick and grab an individual before

it disappears into the school. Flocks of birds fly in formation to utilize the flapping of the front bird's wing, which creates uplift and eases locomotion for the remaining flock. Control in these three natural systems is entirely distributed among the individuals, without having a leader coordinating activities. These natural systems accomplish complex global tasks through simple local interactions of large groups of autonomous individuals and are commonly referred to as examples of swarm intelligence.

Much research in swarm robotics has been conducted *via* computer simulations. Brambilla et al. analyzed more than 60 publications that dealt with swarm robotic collective behaviors in 2013. They found that more than half of these publications presented results which were obtained through simulations or models (Brambilla et al., 2013). Although simulators are a valuable tool for systematically exploring the algorithmic-behavior of a swarm, they frequently involve simplifications and reductionist axioms to enable computational tractability. Such simulated systems can fail to faithfully reproduce the intricate physical interactions and variability that exist in real systems, and their fidelity to the real world is difficult to verify or improve without feedback from physical experiments (Rubenstein et al., 2014).

Building physical systems, however, is a challenging task. Swarm robotics researchers frequently face a cost-functionality optimization problem when it comes to building a scalable robot swarm. For example, every additional sensor on a robot increases the power consumption of the system, requires an additional sensor specific input on the microcontroller, requires additional space, and increases the overall cost. As a result, research in large-scale swarms (>1,000) often sacrifices sensing, processing and locomotion capabilities for the size and quantity of robots, and these design decisions substantially limit the type of experiments that researchers can perform. Instead of increasing the quantity of robots in a swarm by reducing the functionality of each robot, the robotics community requires new technologies that deliver increased functionality at low cost.

Motivation

Our work on technologies for swarm robotics is motivated by three primary objectives, we want to: decrease the cost of fabrication, ease the process of fabrication, and increase the precision of locomotion. We believe that these three factors, among many others, play a crucial role in the development of the next generation of swarm robotic systems. In addition to the obvious focus on decreasing cost, we observed that there is a considerable manufacturing-assembly overhead for existing swarm systems that use either wheeled or slip-stick locomotion. Every component on a robotic system that has to be manually assembled by the researcher invokes a labor cost. This requirement for manual labor by skilled-engineers limits the practicality of fabricating and experimenting with robot collectives at scale.

Improving movement precision enhances localization, whereas precise localization is a *useful* technology to achieve coordination and control of swarm robots (Wu et al., 2014). It is not an easy task for simple robots to maneuver precisely and to reach a common goal. Generally, the difficulties are due to hardware constraints such as small sensor ranges, very limited computational power, little memory, and imprecise locomotion (Moeslinger et al., 2011). For example, low-cost locomotion strategies such as slip-stick locomotion suffer from imprecise movement. Vibration motors provide noisy locomotion without positional feedback, thus preventing a single robot from traveling long distances with any known precision (Rubenstein et al., 2014).

We have developed a locomotion strategy—active low-friction locomotion—that allows agents to maneuver precisely on a discrete two-dimensional grid. Its first embodiment—the HoverBot system—is easy to fabricate and to further-customize. The entire robot consists of a single printed circuit board (PCB), surface-mount components, and a battery. HoverBots can be ordered in large-number from a circuit-board manufacturer in panel-format and arrive fully populated with components—ready to use—thereby lowering the barrier to entry for researchers wishing to study complex systems using swarm robots.

Locomotion Strategies of Swarm Robotic Systems

This study briefly reviews locomotion strategies used by previous swarm robotic systems, it introduces our new locomotion strategy, and puts it into perspective against the literature. Specifically:

(1) We analyze the locomotion methods of 16 swarm robotic systems found in the literature and provide a summary in **Table 1**. The content of **Table 1** is based on the cited work shown in the first column of each row.
(2) We associate each locomotion strategy (wheeled and slip-stick locomotion) with a representative system from **Table 1**. We compare, in detail, the advantages and disadvantages of locomotion strategies in **Table 2**.
(3) We explain and demonstrate our active low-friction locomotion strategy, and we present its first implementation, the HoverBot system.

Tables 1 and **2** contain specific terminology. While most terminology for these features is self-evident, we provide here a summary for those that may be unclear. "Hardware odometry" is defined as the use of sensors to estimate change in position over time. This term indicates systems which do not possess a real form of odometry or which address the lack of hardware odometry by performing collective algorithms such as in Rubenstein et al. (2012). In this column, N/A refers to the fact that the cited publication does not explicitly state information about odometry. "Type of motion" clarifies whether a motion is continuous or discrete and if discrete with what step size. "Dependencies" refer to specific environments which the robots require to function properly. "Surface-mount-technology (SMD)" components are components which can be soldered directly onto a PCB. "Non-SMD" components are usually incompatible with pick-and-place machines and often require manual assembly which generally increases the labor effort and cost for mass manufacture.

Previous Swarm Robotic Systems

The swarm robotic systems listed in **Table 1** use either wheeled or slip-stick locomotion. Slip-stick locomotion refers to the alternation between slipping and sticking of an agent to a substrate that results into directed locomotion (Vartholomeos and

TABLE 1 | Comparison of 16 swarm robotic systems found in the literature.

Reference	Robot name	Locomotion strategy	Propulsion system	Hardware odometry
Mondada et al. (1994)	Khepera	Wheeled	DC motors	Wheel encoders
Siegwart and Caprari (2003)	Alice	Wheeled	Bidirect. motors	Wheel encoders
Mondada et al. (2003)	S-bot	Treeled	DC motors	Wheel encoders
Kornienko et al. (2005)	Jasmine	Treeled	DC motors	N/A
Seyfried et al. (2005)	I-swarm	Slip-stick	Piezoelectric Polymer	No
Mondada et al. (2006)	e-puck	Wheeled	Stepper motors	Wheel encoders
Turgut et al. (2007)	Kobot	Wheeled	DC motors	N/A
Bonani et al. (2010)	MarXbot	Treeled	Rotational motors	Accelerometer, gyroscope
Rubenstein et al. (2012)	**Kilobot**	**Slip-stick**	**Vibration motors**	**No**
McLurkin et al. (2012)	R-one	Wheeled	DC motors	Wheel encoders
Riedo et al. (2013)	Thymio II	Wheeled	DC motors	N/A
Farrow et al. (2014)	Droplet	Slip-stick	Vibration motors	No
GCtronic (2017)	Elisa-3	Wheeled	DC motors	Wheel encoders
Pickem et al. (2015)	**GRITSBot**	**Wheeled**	**Stepper motor**	**Stepper motor**
Wilson et al. (2016)	Pheeno	Wheeled	DC motors	Wheel encoders
Nemitz et al. (2017) (this paper)	**HoverBot**	**Low-friction**	**Planar coils**	**Hall-effect sensor**

*The three highlighted rows depict the swarm robotic systems whose locomotion strategy is further analyzed in **Table 2**.*

TABLE 2 | Comparison of wheeled, slip-stick, and low-friction locomotion.

Feature	**Wheeled** (Pickem et al., 2015)	**Slip-stick** (Rubenstein et al., 2012)	**Active low-friction**
Robot velocity (cm/s)	25	1	0.64
Type of motion	Continuous	Continuous	Discrete—equidistant ~20 mm steps[a]
Battery lifetime	30 min–5 h at 150 mAh	3–24 h at 160 mAh	25 min–600 h at 300 mAh
Dependencies	No	Flat surface	Levitation table with embedded magnets
Hardware odometry	Stepper motors	No	Hall-effect sensor
Actuator calibration	Not required	Required	Not required
Number of non-surface-mount components[b]	≥4	≥5	0
Difficulty of mechanical assembly[c]	(1)(2)(3)(4)(5)(6)	(1)(3)(6)	(6)
Cost[d] at 1,000 units ($)	13.34	3.12	1.96

[a]*Robust (error tolerant) movement on a discrete grid is equivalent to precise movement.*
[b]*Wheeled: two wheels, two motors, and motor control board. Slip-stick: three legs, two vibration motors, and electronics. Active low-friction: electronics.*
[c]*(1) Soldering non-surface-mount components, (2) cutting components, (3) gluing components, (4) screwing components, (5) stacking components, and (6) connecting battery.*
[d]*Cost for components that are solely associated with locomotion, in order quantities of 1,000.*

Papadopoulos, 2006). The vast majority of swarm robotic systems use wheeled locomotion with DC motors and wheel encoders. There are a few exceptions which use tracks and wheels (treels) and accelerometers, gyroscopes, or stepper motors for odometry. Treels are considered as wheeled locomotion. Three systems use slip-stick locomotion, whereas two of those three systems use vibration motors and the remaining system uses piezoelectric polymers as actuators. The HoverBot System is the first implementation of our active low-friction locomotion.

Comparison of Locomotion Strategies

Table 2 compares wheeled, slip-stick, and low-friction locomotion by using the GRITSBot, the Kilobot, and the HoverBot as representative systems. We selected Kilobot as a representative for slip-stick locomotion because it is the first and only large-scale robot swarm exceeding a collective size of 1,000 units. We chose GRITSBot as a representative for wheeled locomotion. Pickem et al. (2015) have presented a recent system that explores both cost and functionality.

While wheeled locomotion has advantages in robot velocity, platform independence, hardware odometry, and actuator calibration, it has disadvantages in battery lifetime, number of non-SMD components (minimum two wheels and two motors), difficulty of mechanical assembly, and cost (including motor control board). In Pickem et al.'s work, non-surface-mount components had to be soldered, receiver coil wires needed to be cut and glued, wheels had to be screwed onto motors, circuit boards needed to be stacked, and the battery had to be connected.

In comparison, slip-stick locomotion has advantages in battery lifetime and cost, but disadvantages in robot velocity, the dependency on flat surfaces, hardware odometry, actuator calibration, and number of non-SMD components (the minimum number being three legs and two vibration motors). In Rubenstein et al.'s work, their mechanical assembly consisted of soldering non-surface-mount components, gluing vibration motors to the robot, and connecting a battery.

Our active low-friction locomotion has advantages in that it provides hardware odometry, requires no actuator calibration, has no non-SMD components, simple mechanical assembly, and is low cost; but it has disadvantages in robot velocity, dependency on a levitation-magnet table, and battery lifetime. To mechanically assemble our robot, one must only connect a battery.

Overall, each of the three strategies possesses specific advantages over the others.

The contribution of this study is the introduction of an active low-friction locomotion mechanism and its first embodiment, the HoverBot system. In addition to using active low-friction locomotion, the HoverBots have the following characteristics, they:

- possess odometry by using a single Hall-effect sensor;
- only require electronic components that are surface mountable;
- only require connecting a battery to a robot as an assembly step;
- use low-cost actuators and associated circuitry;
- do not require actuator calibration;
- move precisely on a discrete grid.

LOW-FRICTION LOCOMOTION

To move—on land, in water, or in the air—always requires an expenditure of energy. Reducing the resistance to motion, namely, friction, allows a greater range of travel for a given input of energy (Radhakrishnan, 1998). However, instead of enhancing locomotion, we enable locomotion by reducing friction. A good example of our proposed locomotion mechanism can be observed in nature. *Nannosquilla decemspinosa* is a small stomatopod found in sand substrates on the Pacific coast of Central and South America. These stomatopods are capable of maneuvering if supported by a 1-mm layer of water and lose this capability once their surrounding dries up (Caldwell, 1979).

The HoverBot is conceptually similar to *N. decemspinosa* and is only capable of maneuvering if it is supplied with a constant air flow beneath its contact surface. The airflow reduces the friction between robot and table allowing relatively weak forces to be used for locomotion. Specifically, we embedded permanent magnets into a levitation table. The HoverBot possesses planar coils which interact with these permanent magnets, resulting in two-dimensional locomotion. Such forces would be insufficient if friction had not been reduced. This concept relaxes actuator boundaries allowing a significant simplification of the robot's actuation and control system.

We define *active* low-friction locomotion as a locomotion type that enables robots to maneuver autonomously, and we define *passive* low-friction locomotion as locomotion type that allows robots to maneuver heteronomously.

Not included in **Table 1**, but relevant to our technical approach, is work from Groß et al. (2011), Napp et al. (2011), Cappelleri et al. (2014), and Pelrine et al. (2017). Groß et al. reported on an experimental setup in which they investigated aided assembly with floating building blocks using an air table. Their system used *passive* low-friction locomotion in which their building blocks did not possess locomotion capabilities, but modules would flow passively in the agitated medium. Napp et al. investigated stochastic interactions between active and passive robots using *passive* low-friction locomotion. Passive robots were foam blocks with complementary shape and embedded magnets that assembled over time on an air bed. Active robots, while not capable of autonomous movement, could expend energy to disassemble the passive robots. Cappelleri et al. introduced a novel approach to achieving independent control of multiple robot magnets. In their work, they designed a grid of planar microcoils. The coils were used to generate magnetic potentials to control the trajectories of magnets. Pelrine et al.'s work is similar to Cappelleri's, but differs in that they add onto their PCBs a thin graphite layer that makes their magnet robots levitate. Both their work feed into additive micromanufacturing with swarms. Similarly to Groß's and Napp's work, agents did not possess locomotive autonomy but were moved by external stimuli; all four approaches are relevant but distinctly different to the work we present here.

The Levitation–Magnet Table

Figure 1 illustrates the concept, and our implementation, of the levitation–magnet table. The table supplies an airflow beneath the HoverBots' contact surface creating an air cushion that reduces friction between the robot and the locomotion substrate. The differential pressure required to lift a HoverBot can be estimated according to Leal (2007) by the following equation:

$$\Delta P = (P_2)_{\min} - \left(P_{\text{amb}}\right) \geq \frac{M * g}{\pi * R^2}. \tag{1}$$

Equation 1 implies that an increase in the robot's weight or a reduction of the robot's surface area can be encountered by an increase in differential pressure. In our experiments, we required approximately 22.5 mm H_2O differential pressure to levitate HoverBots. We measured the differential air pressure between air chamber and ambient environment by using a u-tube manometer. We controlled the air blower's supply voltage with an adjustable transformer (Variac) which varied the air blower's output air-flow-rate, and which in-turn varied the differential pressure between the inside and outside of the levitation table.

The levitation–magnet table measures 200 mm × 300 mm and has an array of permanent magnets embedded into its surface. The permanent magnets serve a double purpose, they: (1) act as magnetic anchors that a HoverBot utilizes to maneuver and (2) give rise to a magnetic field with a discrete regular pattern of features which a HoverBot with a Hall-effect sensor can utilize for odometry. All magnets were assembled mono-directionally: north-pole facing up.

The HoverBot

A HoverBot consists of a single four-layer PCB, shown in **Figure 2**, and a detachable 300 mAh lithium polymer battery. The bottom layer comprises five planar actuation coils. Each HoverBot has a diameter of 39 mm and weighs 19.4 g with, and 7.4 g without, a battery. HoverBot possesses a low-power microcontroller, programming and debug ports, an infrared transceiver, a Hall-effect sensor, and a transistor circuit.

Actuation

We embedded the planar coils in a cross-formation into the bottom layer of the PCB. Each actuation coil has 17 turns and a trace width of 150 μm. A trace width of 150 μm and one $\frac{oz}{ft^2}$ trace thickness allows maximum currents of approximately

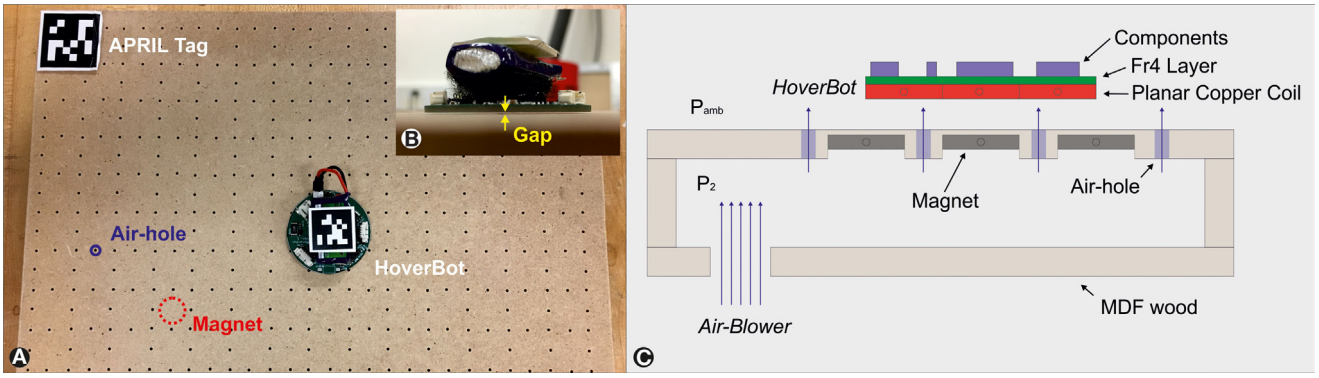


FIGURE 1 | **(A)** Levitation–magnet table with a HoverBot on top. There are four AprilTags for each of the table corners (one visible in the figure) and one AprilTag attached to the HoverBot. The AprilTags are used for tracking (Olson, 2011). **(B)** A photograph showing an air gap between a HoverBot and the table. **(C)** Conceptual overview: an air blower increases pressure P2 within the air chamber. The pressure difference between Pamb and P2 causes a HoverBot to levitate, hence the friction between robot and table decreases.

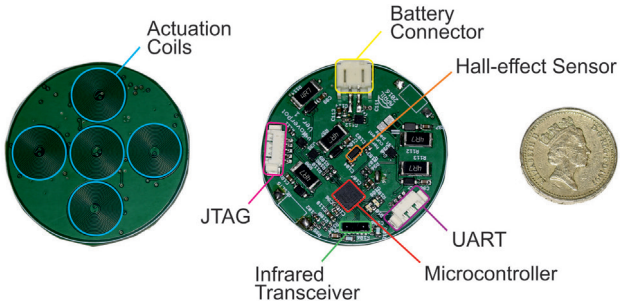


FIGURE 2 | A HoverBot. The bottom layer of the HoverBot consists of an array of five planar actuation coils. Its top layer is populated with Hall-effect and infrared sensors and a low-power (SAMD21E) microcontroller. The battery of a HoverBot is detached in this figure.

300 mA based on the Generic Standard on the Printed Board Design (IPC-2221) charts. We set the maximum current per coil to 500 mA, which induces a magnetic field of 1.1 mT. Our design uses a maximum current that is greater than the suggested standard, because we decided to evaluate the circuitry to its upper limits. We measured the magnetic field by using an InvenSense MPU-9250 magnetometer. We placed the magnetometer onto the core of the center coil.

Each coil is connected in series with a current limiting resistor and a transistor. If the transistor switches on, a constant voltage is applied across the coil and resistor. The transistor's switching behavior is controlled by a pulse-width-modulated (PWM) signal from the microcontroller. We control the amount of current through the coil by changing the duty cycle of the PWM. The magnetic field of a solenoid can be approximated by Ampere's law:

$$B = \mu * n * I, \tag{2}$$

$$n = \frac{N}{L}, \tag{3}$$

where B is magnetic flux density, μ is permeability; n is turn density; I is current; N is the number of turns; and L is unit length.

Fundamentally similar to Eqs 2–3, the magnetic field of a planar coil is dependent on the coil's turn density and the current flow. The number of turns is identical for every HoverBot. However, the coil, trace, and current limiting resistor resistance could vary due to manufacturing tolerances and cause a change of current flow for a given duty cycle. We measured the average series resistance of 15 actuation circuits of a total of 3 HoverBots with a Fluke 115 multimeter. The SD was 0.1 Ohm, which causes a current change of 7 mA. Hence, the potential current fluctuations are less than 1.5% and can be neglected. HoverBots do not require any kind of actuator calibration.

Sensing and Communication

A HoverBot possesses infrared and Hall-effect sensors. The Hall-effect sensor can be used for odometry and the detection of local magnetic fields. The infrared transceiver can be used for robot-to-computer communication. Our current HoverBot version does not allow robot-to-robot communication due to limitations in its hardware configuration. It only possesses a single IR transceiver pointing upwards.

Programming and Debugging

A HoverBot has programming and debug ports (IR transceiver, JTAG and UART). We programmed the HoverBot *via* JTAG using an Atmel SAM-ICE programmer. Therefore, this HoverBot version requires a wired connection to be programmed. We debugged HoverBot *via* infrared using an infrared handheld device.

Power System

A HoverBot has 3.7 V 300 mAh lithium polymer batteries attached to it. We calculated the minimum battery life by accumulating the currents that occur during locomotion.

The current locomotion strategy requires a constant current of approximately 720 mA, which allows a minimum battery life of around 25 min. However, lithium polymer batteries should never be completely discharged due to their chemistry. We wrote a battery-watch program to monitor the battery during runtime, and this program shuts down all circuitry when the battery reaches 90% depletion. The maximum battery life is calculated by considering HoverBot when in sleep mode, in which it approximately consumes 500 μA. In this low-power mode, the HoverBot's battery life time rises to around 600 h or 25 days. In this HoverBot version, the lithium polymer batteries have to be detached for charging. We charged the batteries by using a Turnigy Micro-6 LiPoly battery charger.

Locomotion Strategy

The HoverBot levitates on air cushions and maneuvers by sequentially energizing its planar coils to pull itself toward magnetic anchors. **Figure 3** indicates a HoverBot's open-loop locomotion strategy. A single step, a movement from one magnetic anchor to another, is decomposed into three part steps. In step 1, HoverBot starts from its idle state in which its center coil is aligned with a magnet, and the other four coils are each overlapping with adjacent magnets. The HoverBot simultaneously actuates one side coil with maximum current and the opposite side coil with medium current. This actuation results in an overall movement to the right while preventing HoverBot from rotating. Subsequent steps are conceptually the same, but each step requires a differing pair of coils to be actuated. Three of these steps are required for a HoverBot to move from one magnet to another. This actuation scheme only enables complete magnet-to-magnet movements. A change of direction during a part step has not been investigated. The relative positions of the HoverBot coils and the magnets are crucial for this actuation scheme. We chose magnet-to-magnet and coil-to-coil pitches based on Eq. 4 to ensure a 50% overlap between actuator coil and an adjacent magnet at any given step assuming that coil and magnet diameters match. Therefore, HoverBot's minimum step size is the pitch between adjacent magnets (2 cm pitch).

$$r_{\mathrm{mc,c}} = \frac{d_{\mathrm{mc}}}{d_{\mathrm{c}}} = \frac{d_{\mathrm{m}} - d_{\mathrm{c}}}{d_{\mathrm{c}}} = \frac{1}{2}, \tag{4}$$

where d_{m} is magnet to magnet pitch; d_{c} is coil to coil pitch; d_{mc} is magnet to coil pitch; and $r_{\mathrm{mc,c}}$ is ratio of d_{mc} to d_{c}.

A HoverBot moves in a two-dimensional discrete environment. The programmer cannot deliberately rotate a HoverBot or move it in any other trajectories than the Manhattan Geometry.

Odometry

Figure 4 is based on the Hall-effect sensor readings from a HoverBot during movement which were paired with spatial information from the AprilTags. While a HoverBot moves from one magnetic anchor to another (2 cm pitch), its Hall-effect sensor measures a continuously changing magnetic flux density as indicated in **Figure 4**. The Hall-effect sensor is centered above the center coil and is capable of measuring magnetic flux densities from −73 to +73 mT. The maximum readings occur when the HoverBot's center coil is aligned with a magnetic anchor. Although our current actuation scheme operates as an open-loop control, the magnetic flux density changes over distance depict distinct features in the two-dimensional space which could be utilized as feedback for closed-loop control. We have not experienced distorted sensor (Hall-effect, IR) readings due to magnetic interference.

Manufacture and Cost

The circuitry of a HoverBot only consists of surface-mount components as indicated in **Table 2**. HoverBot is designed explicitly for manufacturability; it consists of a single PCB and therefore mass manufacture is a simple case of placing a batch order with a PCB foundry. HoverBots can be autonomously populated with pick-and-place machines at the point of manufacture. Assembly of one robot takes seconds since it only consists of plugging in a

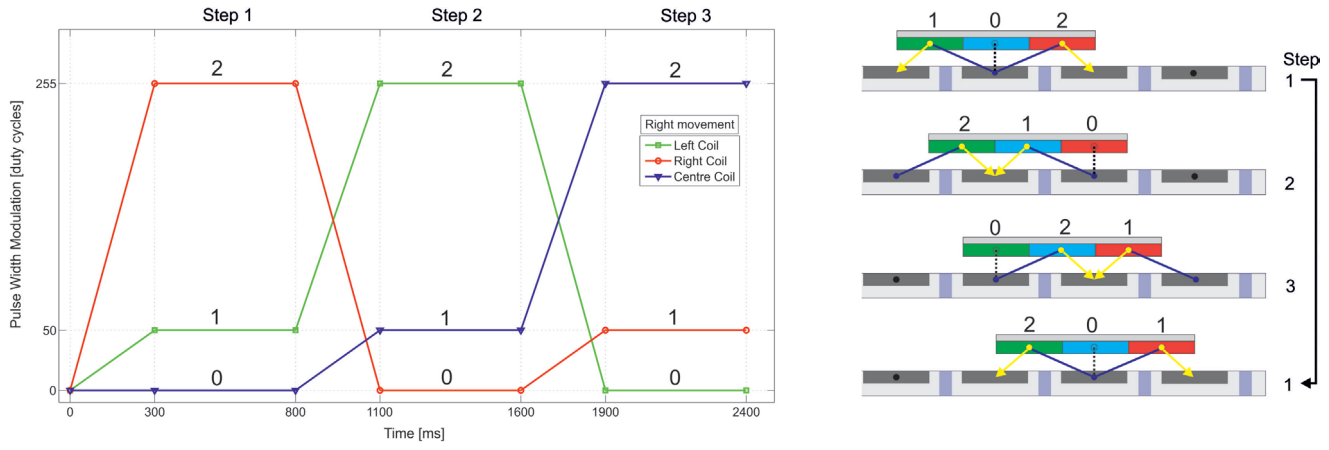


FIGURE 3 | The illustrated actuation scheme describes a movement to the right. The other directions can be derived from this actuation profile. A HoverBot moves from one magnetic anchor to another by performing three steps. Starting in idle state (step 1), a HoverBot always actuates two coils at the time for any given step since we require two points in space to maintain the orientation of a plain. Therefore, this actuation sequence prevents rotation during movement.

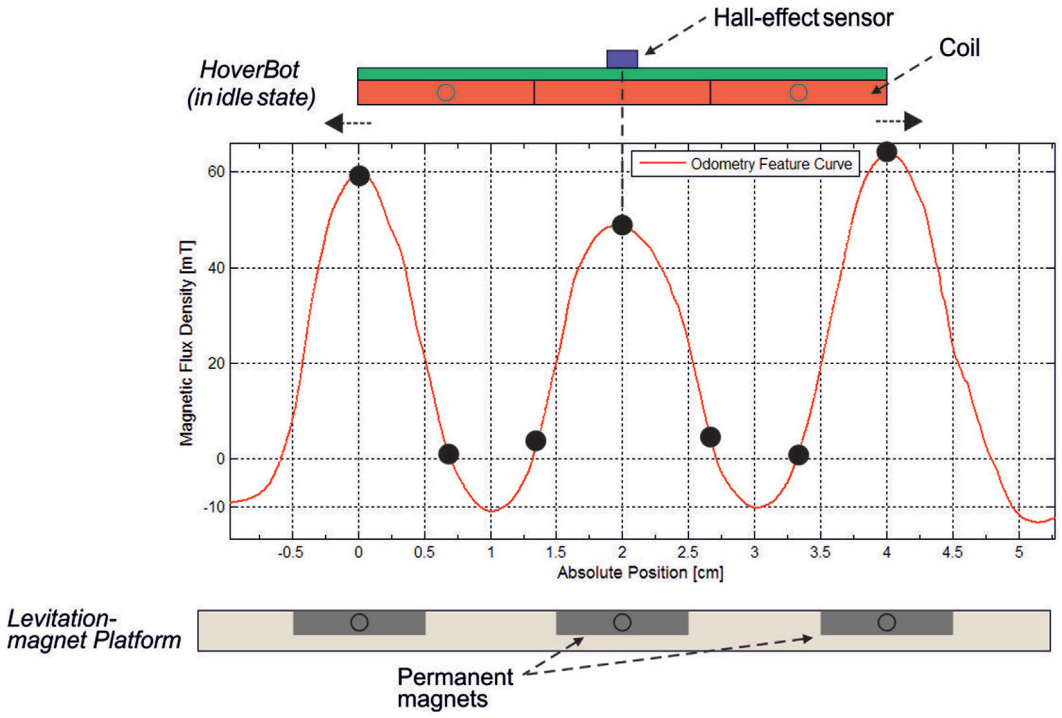


FIGURE 4 | Experimental results showing that the magnetic flux density peaks correspond to the centers of the permanent magnets. The magnets are 2 cm displaced from one another. A HoverBot in idle state aligns with its Hall-effect sensor above a magnet. A step from one magnet to another causes the Hall-effect sensor to measure a peak-to-peak magnetic field curve. Specifically, a movement is decomposed into three individual steps in which the Hall-effect sensor measures a distinct magnetic flux density after each step, as indicated by a black dot.

TABLE 3 | Cost summary for HoverBots in order quantities of 15 units, and for one levitation–magnet table.

Levitation–magnet table		HoverBots at 15 U	
Category	**Cost ($)**	**Category**	**Cost ($)**
Wood	20	Locomotion	4.10
Air Blower	55	Computation	2.05
Variac	135	Power	4.96
Magnets (140)	25	Sensing	1.09
		Communication	4.28
		Miscellaneous	3.23
		Printed circuit board manufacture	2.66
Total	235	Total	22.37

battery to a HoverBot, also indicated in **Table 2**. The fabrication of the levitation–magnet table is described in detail in Section "Fabrication."

Table 3 summarizes the costs of the current levitation–magnet table and HoverBots in order quantities of 15. The levitation–magnet table costs $235 whereas each HoverBot costs $22.37 in quantities of 15 and $11.88 in quantities of 1,000 s. The most expensive part of the levitation–magnet table is the variable transformer at $135. The costs for components that are solely associated with HoverBot's actuation system (transistors, shunt resistors, diodes, and capacitors) are $1.96 in order quantities of 1,000 as indicated in **Table 2**.

The bill of materials, HoverBot system design files, and code are available on request.

EVALUATION

To evaluate the HoverBot system we designed a controllable experimental setup. We used artificial features (fiducials)—AprilTags (Olson, 2011)—which we placed on top of the HoverBot and at each corner of the table. AprilTags are robust to occlusions and lens distortion while being very efficient in achieving detection rates of 20 Hz in our setting. To measure the accuracy of the HoverBot, we tracked the centroid and the orientation of the robot by detecting the corresponding AprilTag. **Figure 5** depicts the main features of the tracking system. This system can run for hundreds of minutes without human intervention, thereby automating the data acquisition pipeline. We used a Chameleon 1.3 MP Color (Sony IXC445) camera and a Tamron 13FM28IR 2.8 mm f/1.2 day/night lens.

We tested the HoverBot system and its low-friction locomotion by conducting eleven experiments that lasted a total of 3 h and more than 10,000 steps. In these experiments, the HoverBot circled on an arbitrary trajectory until it was nearly discharged. We used a set of AprilTags to track the HoverBot over time and subsequently evaluated its distance traveled, velocity, and number of missteps (errors). With our current actuation sequence, the HoverBot moves an average of 0.64 cm/s with an SD of

0.015 cm/s. We did not observe any missteps or accidental rotations during these three hours. A "misstep" is defined as an unsuccessful series of energized coils that results in the robot staying on its previous position. An "accidental rotation" is defined as an inadvertent robot rotation by 45° due to local table imperfections (e.g., air flow fluctuations) or collisions with other robots or static objects. A video recording of this experiment is provided by the Supplemental Video—SV1.[1]

Although the HoverBot moves robustly, we observed unintentional shaking in all four directions during movement. Therefore, we compared moved distance, which includes the total distance including unintentional shaking, with effective distance, which is the actual distance between waypoints. We define ε by the following term:

$$\epsilon = \frac{\text{moved distance}}{\text{effective distance}}. \quad (5)$$

We found that ε is 2.29 (on average) with an SD of 0.27. An ε of 2.29 explicitly states that the HoverBot moves more than two times the distance it travels. ε is directly related to the actuation scheme. We chose an actuation scheme that is relatively slow, but very robust by performing zero missteps over three hours of experiments. ε can be further reduced by changing the actuation scheme and specifically the timing and amount of current that flows through up to five coils simultaneously.

[1]http://edin.ac/2wxEE5w.

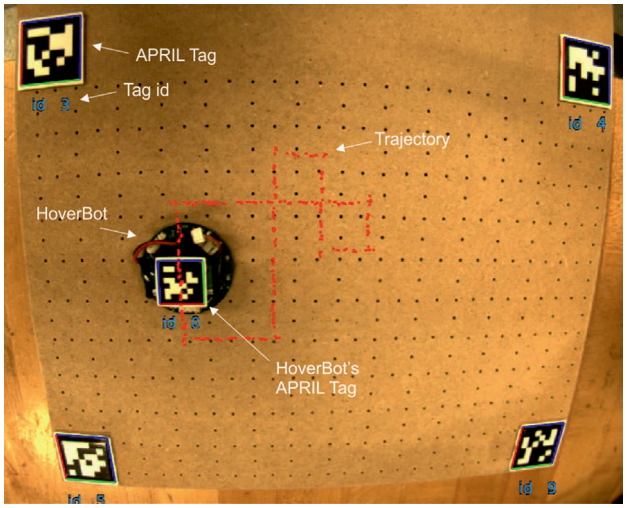


FIGURE 5 | Experimental setup to evaluate a HoverBot's locomotion performance. We placed one AprilTag in each corner of the levitation–magnet table. These tags serve as reference points and allow determination of a HoverBot's relative position over time. Each AprilTag corresponds to an ID number. During experiments, we read out each AprilTag's ID, x-position, y-position, and rotation. The red line indicates a HoverBot's trajectory, which reinforces with each lap.

Recovery from a Locked Rotational Position

Although we have not experienced any accidental rotation incidents during three hours of testing, we developed an actuation strategy that allows a HoverBot to recover from a locked position. As shown by **Figure 4**, the Hall-effect sensor measures a local magnetic minimum if a HoverBot is locked due to accidental rotation. When a HoverBot recognizes this state, it can execute a recovery actuation scheme. It first actuates only its center coil to change from the position in **Figure 6A** to the position in **Figure 6B**. Then it actuates, in addition, a side coil to regain the correct orientation as indicated in **Figure 6C**. We recorded this sequence, a video–recording of this experiment is provided by the Supplemental Video—SV2.[2]

DEMONSTRATION

In addition to our quantitative evaluation of HoverBot's locomotion capabilities, we performed four additional demonstrations to give more insights into the nature of the HoverBot

[2]http://edin.ac/2wcISwJ.

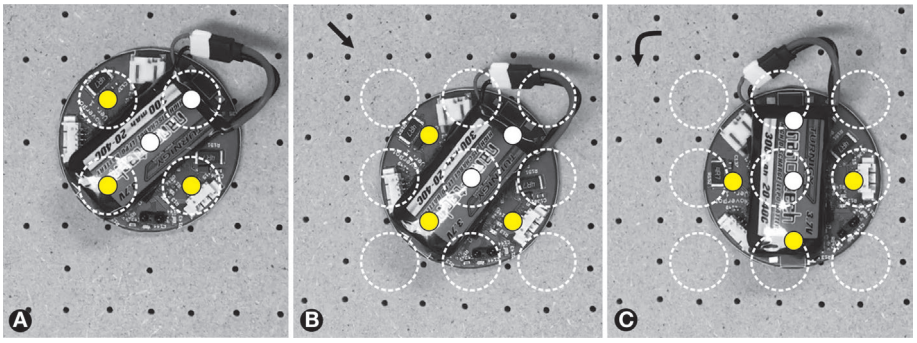


FIGURE 6 | Recovery in locked position. **(A)** A HoverBot is locked in a 45°-angled position. Four of its five coils are aligned with permanent magnets. **(B)** The HoverBot rapidly pulsed (only) its center coil and regained center coil alignment with a permanent magnet. **(C)** The HoverBot additionally actuated a side coil and regained a slightly shifted idle position. At this stage, the HoverBot is reenabled to move.

system. **Figure 7** indicates two HoverBots moving in formation (A), moving randomly, colliding, and recovering from rotation (B), colliding (C), and colliding while one robot is in sleep mode, acting as a passive agent (D).

We observed that two HoverBots that move independently in formation become unsynchronized over time due to oscillator imperfections. Physical inter-robot interactions can either result in robots maintaining their position after collision, which likely happens in a frontal collision event, or *robots* loose orientation and have to recover. The random collision demonstration also indicates possible orientation loss due to rapid and constant change in direction. Those incidents, however, are scarce, they were detected, and were recovered from. In most cases, moving HoverBots are capable of pushing passive agents, sliding them to one side, or pushing them in front, in the direction of travel. However, we recorded one incident in which a moving agent could not pass a passive agent due to a specific physical orientation. Video recordings of the experiments which correspond to **Figures 7A–D** are provided by the Supplemental Videos—SV3–SV6.[3]

DISCUSSION

Battery Life and Robot Velocity

HoverBot possesses a relatively short battery lifetime (~25 min) due to high coil actuation currents that are required to achieve magnetic fields of approximately 1.1 mT. According to Eqs 2 and 3, the magnetic field is linearly dependent on the actuation current, but also on the number of coil windings. An increase of coil windings as well as the stacking of planar coils (multilayer PCBs) could significantly decrease the power consumption.

The existing robot velocity can be improved without an increase of power consumption. The product of current and time for slow coil actuation does not change for rapid coil actuation. HoverBot's velocity is currently slow because we wanted to start off with a robust actuation scheme. Future work will have to investigate faster actuation schemes. It is very likely that actuator calibration will become necessary once we reach HoverBot's physical speed limits. The actuation schemes will become more delicate and have to energize the actuation coils extremely precisely, both in terms of the amount, duration, and direction of current flows. One solution to this control problem could be to use machine learning algorithms. An external camera system could send feedback to the HoverBot agent and inform the controller whether movement was successful or not.

[3]SV3: Formation: http://edin.ac/2wxt0aN, SV4: Random Collision: http://edin.ac/2wdsDzt, SV5: Collision (active): http://edin.ac/2wwTTeJ, SV6: Collision (passive): http://edin.ac/2wd3h4Y.

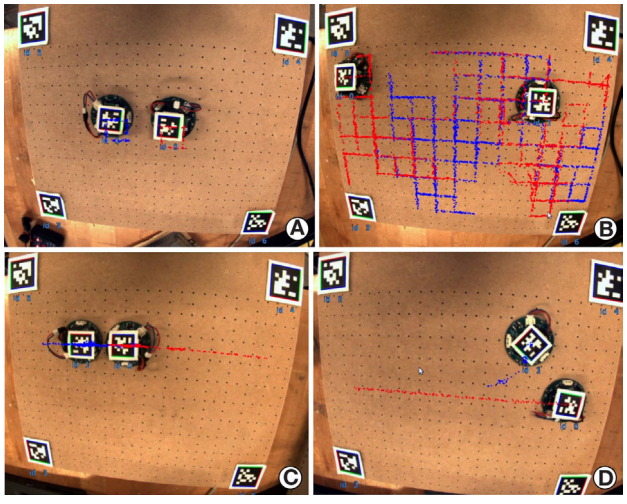


FIGURE 7 | Demonstrations of the locomotion capabilities of multiple HoverBots. **(A)** Two HoverBots circle in formation until they are unsynchronized—video SV1. **(B)** Two HoverBots move randomly, collide, and recover—video SV2. **(C)** Two HoverBots collide frontally with one another—video SV3. **(D)** One HoverBot collides with a passive HoverBot—video SV4. Red and blue trajectories depict the HoverBot's movements over time.

Ease of Robot Fabrication

Although HoverBots only consist of surface-mount components, we believe the importance of this advantage will decrease over time. The current state of swarm robotics research requires low-cost, easy-to-fabricate, and easy-to-use swarm robotic systems. However, once we obtain a better understanding of complex systems and how emergence occurs, cost and ease of fabrication will become secondary because the risk-factors involved in deploying swarms (system failure, loss of control, and safe and reliable operation) will have decreased. Furthermore, there are many great examples in industry in which very sophisticated products have been mass manufactured (computers, cars, airplanes, etc.). Investing into an expensive swarm of robots will become worthwhile once we know how to safely operate and control it.

The Table

The existing ratio of magnet-to-magnet and coil-to-coil distances was chosen to simplify HoverBot's actuation circuitry by only requiring coils to be energized in one direction. In future work, we can investigate the use of H-bridge drivers to improve locomotion by allowing bidirectional currents to energize the actuator coils. There may also be a benefit of designing different magnet patterns, such as those which that vary between polarities as well as exploit different geometric constellations (e.g., concentric patterns).

Scaling the System

The current table measures 200 mm × 300 mm, and this size limits the maximum number of robots on the table to 35, assuming a lattice robot formation without extra space for movement and a robot diameter of 40 mm. There is no reason why the table or robot could not be scaled in either direction. The table size could be significantly increased, to the size of an air hockey table for example. The differential pressure that causes the robots to levitate can be easily increased by using a more powerful blower, or even several at once. An increase in differential pressure would allow greater payloads to be carried by the robots. The robot size

could be significantly increased or decreased. There are micromachining systems that are capable of fabricating 50 μm wide copper traces (e.g., LPKF Protolaser U3) allowing much smaller actuator coil sizes. The 300 mAh battery could be substituted with less powerful batteries or even replaced with solar panels.

Future Directions

HoverBot version-2 should possess four directional communications to increase further its utility as swarm algorithmic testbed.

The collision of an active with a passive robot in video SV6 indicate an opportunity for new swarm robotic algorithms in which passive and active robots are being utilized to achieve a task. A passive robot might become active if it has not been pushed around by another robot for a defined period of time. A passive robot might also specialize in sensing and inform active robots about its observations. This heterogeneity might lead to strategies that optimize the power budget of the swarm while solving the task at hand.

The formation demonstration in video SV3 indicates that the HoverBot system can be used for even larger collective movements. This behavior is difficult to achieve with wheeled or slip-stick actuated swarm systems since such systems move in continuous space and must rotate to change directions. HoverBots locomotion can be compared with that of quadrotors in formation flight (Kushleyev et al., 2013), maintaining orientation of the local and global directions.

Almost all of HoverBot's advantages originate from its minimalist design. HoverBots levitate, move precisely on a discrete grid, and are capable of verifying a step by measuring continuously magnetic flux densities. We will utilize this combination of discrete motion with continuous local perception to study search and tracking as well as mapping algorithms. An excellent starting point is Senanayake review on search and tracking algorithms for swarm robots (Senanayake et al., 2014).

CONCLUSION

In this study, we introduced a new locomotion strategy, active low-friction locomotion, and showed its first embodiment: the HoverBot system. We demonstrated HoverBot's capabilities by performing six different experiments ranging from moving in a predetermined trajectory, to random movement and inter-robot collisions. Active low-friction locomotion is an alternative to wheeled- and slip-stick locomotion in the field of swarm robotics. The HoverBot system possesses odometry by using a single Hall-effect sensor, it only requires components that are surface mountable, it only requires connecting a battery as assembly step, it uses low-cost actuators and associated circuitry, does not require actuator calibration, and moves precisely on a discrete grid. The HoverBot systems offer a unique combination of discrete precise motion with continuous local perception. Its hardware can be easily extended with additional sensors. Potential research directions using this embodied-simulation system will include search and tracking, or mapping with robot swarms. The HoverBot system serves as a testbed for new hardware and algorithms.

FABRICATION

Fabrication of Levitation–Magnet Table

We purchased 10 mm wide and 3 mm thick cylindrical N42 magnets from Amazon. We bought 12.7 mm thick medium-density fiberboard from a local hardware store. We used a ShopBot Buddy to mill and drill holes. We used a 0.063″ drill bit for the air-holes and a 0.394″ end-mill for the magnet pockets. We placed the top-plate of the air table on an optics (metal) table and embedded the magnets mono-directionally (polarity) into the pockets. We used an Arrow TR400 glue gun to fix the magnets in the pockets. We used a Mcculloch MCB2205 leaf blower as the air source in combination with a Circuit Specialists 16VA520T20 Variac for airflow control. The air blower listed in **Table 3** is the Black & Decker BV5600 High Performance Blower (for price reference) and is equivalent to the MCB2205. We leveled the levitation–magnet table using a water scale.

Fabrication of a HoverBot

We purchased all electronics components from Digikey. The circuit boards were designed with CadSoft Eagle and manufactured by 4PCB.com. We soldered the components by using a hot air pencil and an airbath preheating system from Zephyrtronics.

AUTHOR CONTRIBUTIONS

MN: created the system and is lead author of all sections of the work. MS: contributed building the HoverBot table, developing the experiment scheme, and revised the manuscript. JM: contributed building the HoverBot agent. GF: contributed to the AprilTag setup, data analysis, revising and partly writing the manuscript. LT and RM: contributed to the manuscript/revision and the development of the table. EO and AH: advised on building the HoverBot system. AS: lead advisor and primary editor of the manuscript.

ACKNOWLEDGMENTS

MN thanks Victoria Edwards (PhD student, University of Michigan) for her helpful comments on the manuscript.

FUNDING

MN gratefully acknowledges support from the Centre in Doctoral Training in Intelligent Sensing and Measurement (EP/L016753/1), UK and the Office of Naval Research (N00014-13-1-0217), USA. This work was supported by EPSRC through the Robotarium Capital Equipment (EP/L016834/1).

REFERENCES

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*, Vol. 1. New York, NY: Oxford University Press.

Bonani, M., Longchamp, V., Magnenat, S., Rétornaz, P., Burnier, D., Roulet, G., et al. (2010). "The marXbot, a miniature mobile robot opening new perspectives for the collective-robotic research," in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 – Conference Proceedings*, Taipei, 4187–4193.

Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell.* 7, 1–41. doi:10.1007/s11721-012-0075-2

Caldwell, R. L. (1979). A unique form of locomotion in a stomatopod – backward somersaulting. *Nature* 282, 71–73. doi:10.1038/282071a0

Cappelleri, D., Efthymiou, D., Goswami, A., Vitoroulis, N., and Zavlanos, M. (2014). Towards mobile microrobot swarms for additive micromanufacturing. *Int. J. Adv. Robot. Syst.* 11(9), 150. doi:10.5772/58985

Caprari, G., and Siegwart, R. (2003). "Design and control of the mobile micro robot alice," in *Proceedings of the 2nd International Symposium on Autonomous Minirobots for Research and Edutainment AMiRE 2003* (Brisbane: CITI), 22–32.

Farrow, N., Klingner, J., Reishus, D., and Correll, N. (2014). "Miniature six-channel range and bearing system: algorithm, analysis and experimental validation," in *Proceedings – IEEE International Conference on Robotics and Automation*, Hong Kong, 6180–6185.

GCtronic. (2017). *Elisa-3*. Available at: http://www.gctronic.com/doc/index.php/Elisa-3

Groß, R., Magnenat, S., Küchler, L., Massaras, V., Bonani, M., and Mondada, F. (2011). "Towards an autonomous evolution of non-biological physical organisms," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Budapest, Vol. 5777 LNAI, 173–180.

Kornienko, S., Kornienko, O., and Levi, P. (2005). "Minimalistic approach towards communication and perception in microrobotic swarms," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, Edmonton, 4005–4011.

Kushleyev, A., Mellinger, D., Powers, C., and Kumar, V. (2013). Towards a swarm of agile micro quadrotors. *Auton. Robots* 35, 287–300. doi:10.1007/s10514-013-9349-9

Leal, L. G. (2007). *Advanced Transport Phenomena: Fluid Mechanics and Convective Transport Processes*. Cambridge University Press, 934.

McLurkin, J., Lynch, A. J., Rixner, S., Barr, T. W., Chou, A., Foster, K., et al. (2012). "A low-cost multi-robot system for research, teaching, and outreach," in *Springer Tracts in Advanced Robotics*, Vol. 83 STAR, Lausanne, 597–609.

Moeslinger, C., Schmickl, T., and Crailsheim, K. (2011). "A minimalist flocking algorithm for swarm robots," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 5778 LNAI, Budapest, 375–382.

Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., et al. (2009). "The e-puck, a robot designed for education in engineering," in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, Vol. 1, (IPCB, Instituto Politécnico de Castelo Branco), 59–65. No. LIS-CONF-2009-004.

Mondada, F., Franzi, E., and Ienne, P. (1994). Mobile Robot Miniaturisation: A Tool for Investigation in Control Algorithms. *Exp. Robot.* 501–513.

Mondada, F., Guignard, A., and Bonani, M., Bar, D., Lauria, M., and Floreano, D. (2003). "SWARM-BOT: from concept to implementation," in *Proceedings. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003),* Vol. 2. (IEEE), 1626–1631.

Napp, N., Burden, S., and Klavins, E. (2011). Setpoint regulation for stochastically interacting robots. *Auton. Robots* 30, 57–71. doi:10.1007/s10514-010-9203-2

Olson, E. (2011). "AprilTag: a robust and flexible visual fiducial system," in *Proceedings – IEEE International Conference on Robotics and Automation*, Shanghai, 3400–3407.

Pelrine, R., Hsu, A., Cowan, C., and Wong-Foy, A. (2017). "Multi-agent systems using diamagnetic micro manipulation—from floating swarms to mobile sensors," in *International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)* (Seattle, WA: IEEE), 1–6.

Pickem, D., Lee, M., and Egerstedt, M. (2015). "The GRITSBot in its natural habitat – a multi-robot testbed," in *Proceedings – IEEE International Conference on Robotics and Automation*, 4062–4067.

Radhakrishnan, V. (1998). Locomotion: dealing with friction. *Curr. Sci.* 74, 826–840. doi:10.1073/pnas.95.10.5448

Riedo, F., Chevalier, M., Magnenat, S., and Mondada, F. (2013). "Thymio II, a robot that grows wiser with children," in *Proceedings of IEEE Workshop on Advanced Robotics and Its Social Impacts, ARSO*, Tokyo, 187–193.

Rubenstein, M., Ahler, C., and Nagpal, R. (2012). "Kilobot: a low cost scalable robot system for collective behaviors," in *Proceedings – IEEE International Conference on Robotics and Automation*, St. Paul, MN, 3293–3298.

Rubenstein, M., Cornejo, A., and Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. *Science* 345, 795–799. doi:10.1126/science.1254295

Senanayake, M., Senthooran, I., Barca, J. C., Chung, H., Kamruzzaman, J., and Murshed, M. (2014). Search and tracking algorithms for swarms of robots: a survey. *Rob. Auton. Syst.* 75, 422–434. doi:10.1016/j.robot.2015.08.010

Seyfried, J., Szymanski, M., Bender, N., Estaña, R., Thiel, M., and Wörn, H. (2005). The I-SWARM project: intelligent small world autonomous robots for micro-manipulation. *Swarm Robot.* 3342, 70–83. doi:10.1007/978-3-540-30552-1_7

Turgut, A. E., Goekce, F., Celikkanat, H., Bayindir, L., and Sahin, E. (2007). Kobot: a mobile robot designed specifically for swarm robotics research. *Architecture* 32–42.

Vartholomeos, P., and Papadopoulos, E. (2006). "Analysis, design and control of a planar micro-robot driven by two centripetal-force actuators," in *Proceedings – IEEE International Conference on Robotics and Automation*, Orlando, FL, 649–654.

Wilson, S., Gameros, R., Sheely, M., Lin, M., Dover, K., Gevorkyan, R., et al. (2016). Pheeno, a versatile swarm robotic research and education platform. *IEEE Robot. Autom. Lett.* 1, 884–891. doi:10.1109/LRA.2016.2524987

Wu, H., Qu, S., Xu, D., and Chen, C. (2014). Precise localization and formation control of swarm robots via wireless sensor networks. *Math. Probl. Eng.* 2014:942306. doi:10.1155/2014/942306

Conflict of Interest Statement: No competing financial interests exist. The subject matter in this study forms the basis of patent application GB 1611448.0.

Appendix F

Journal: Multi-Functional Sensing for Swarm Robots Using Time Sequence Classification: HoverBot, an Example

Journal: Frontiers in Robotics and Artificial Intelligence

Research topic: Programming Multi-Robot Systems in the Real World: Methods and Best Practices

Prof. Carlo Pinciroli *editor*
Assistant Professor
Robotics Engineering
Worcester Polytechnic Institute, USA

Dr.-Ing. Heiko Hamann *reviewer*
Professor for Service Robotics
Institute of Computer Engineering
University of Lubeck, Germany

Dr. Yara Khaluf *reviewer*
Post-doctoral Fellow
Department of Information Technology
Ghent University, Belgium

Dr. Eliseo Ferrante *reviewer*
Post-doctoral Fellow
Department of Biology
University of Leuven, Belgium

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 17 May 2018
doi: 10.3389/frobt.2018.00055

Multi-Functional Sensing for Swarm Robots Using Time Sequence Classification: HoverBot, an Example

Markus P. Nemitz [1,2], Ryan J. Marcotte [2], Mohammed E. Sayed [1], Gonzalo Ferrer [2], Alfred O. Hero [2], Edwin Olson [2] and Adam A. Stokes [1]**

[1] School of Engineering, Institute for Integrated Micro and Nano Systems, The University of Edinburgh, Edinburgh, United Kingdom, [2] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, United States

Scaling up robot swarms to collectives of hundreds or even thousands without sacrificing sensing, processing, and locomotion capabilities is a challenging problem. Low-cost robots are potentially scalable, but the majority of existing systems have limited capabilities, and these limitations substantially constrain the type of experiments that could be performed by robotics researchers. Instead of adding functionality by adding more components and therefore increasing the cost, we demonstrate how low-cost hardware can be used beyond its standard functionality. We systematically review 15 swarm robotic systems and analyse their sensing capabilities by applying a general sensor model from the sensing and measurement community. This work is based on the HoverBot system. A HoverBot is a levitating circuit board that manoeuvres by pulling itself towards magnetic anchors that are embedded into the robot arena. We show that HoverBot's magnetic field readouts from its Hall-effect sensor can be associated to successful movement, robot rotation and collision measurands. We build a time series classifier based on these magnetic field readouts. We modify and apply signal processing techniques to enable the online classification of the time-variant magnetic field measurements on HoverBot's low-cost microcontroller. We enabled HoverBot with successful movement, rotation, and collision sensing capabilities by utilising its single Hall-effect sensor. We discuss how our classification method could be applied to other sensors to increase a robot's functionality while retaining its cost.

Keywords: HoverBot, swarm robotics, multi-functional sensing, dynamic time warping, DTW, barycentre averaging, DBA

OPEN ACCESS

Edited by:
Carlo Pinciroli, Worcester Polytechnic Institute, United States

Reviewed by:
Heiko Hamann, Universität zu Lübeck, Germany
Yara Khaluf, Ghent University, Belgium
Eliseo Ferrante, KU Leuven, Belgium

****Correspondence:***
Markus P. Nemitz
M.nemitz@ed.ac.uk
Nemitz@umich.edu
Adam A. Stokes
Adam.stokes@ed.ac.uk

Specialty section:
This article was submitted to Multi-Robot Systems, a section of the journal Frontiers in Robotics and AI

Received: *22 December 2017*
Accepted: *19 April 2018*
Published: *17 May 2018*

Citation:
Nemitz MP, Marcotte RJ, Sayed ME, Ferrer G, Hero AO, Olson E and Stokes AA (2018) Multi-Functional Sensing for Swarm Robots Using Time Sequence Classification: HoverBot, an Example. Front. Robot. AI 5:55. doi: 10.3389/frobt.2018.00055

1. INTRODUCTION

1.1. Swarm Robotics

Swarm robotics is the study of developing and controlling large groups of simple robots. One goal of swarm robotics research is to substitute a few sophisticated robots with many simple robots to gain robustness, flexibility and to circumvent single-robot-failures from resulting in mission abortions (Brambilla et al., 2013). Applications range from space-exploration to finding survivors after large-scale disasters. Much inspiration in this area has been drawn from nature (Bonabeau et al., 1999). Flocks of birds fly in formation and take turns in positioning to maximise the total travelled distance as a collective. Schools of fish cluster together to increase the chances of survival against a visually orientated

predator. Colonies of termites collaborate to build termite mounds with integrated ventilation mechanisms to protect the colony from critical temperatures. All these systems accomplish complex tasks through simple local interactions amongst themselves, collectives of autonomous agents, and are commonly referred to as examples of swarm intelligence. Swarm robotics can be seen as a research area that emerged from the field of swarm intelligence, whereas swarm intelligence depicts a subfield of artificial intelligence. The first swarm robotic system was the Khepera robot in 1994 (Mondada et al., 1994). Since then, many other swarm robotic systems have been built, most of them are listed in **Table 1**. However, much of swarm intelligence research has been conducted via computer simulations. Brambilla et al. analysed more than 60 publications that dealt with swarm robotic collective behaviours. They found that more than half of these publications presented results which were obtained through simulations or models (Brambilla et al., 2013). Although simulators are a valuable tool for exploring, systematically, the algorithmic-behaviour of natural swarms, they frequently involve simplifications and reductionist axioms to enable computational tractability. Such simulated systems can fail to faithfully reproduce the intricate physical interactions and variability that exist in real systems, and their fidelity to the real world is difficult to verify or improve without feedback from physical experiments (Rubenstein et al., 2014).

Over the past two decades, many swarm robotic systems have been developed. They all differ in certain aspects such as power consumption, locomotion strategy, or sensing capability. The sensing capabilities of a robot influence the type of experiments one can perform. While a camera adds more functionality to a robot than an ambient light sensor, each sensor comes at a different cost. We systematically analysed previous swarm robotic systems and found that some systems possess sensors that have been, or could be, used for the detection of multiple signals. For example, an IR transceiver could be used for communication and proximity sensing amongst others. Developing robots that are low-cost and functional is a challenging task, therefore, utilising sensors for the detection of multiple measurands is desirable. We believe that this concept is very important and deserves further evaluation.

TABLE 1 | Comparison of 13 swarm robotic systems' sensing capabilities.

Robot System / Number of Measurands	2	**3**	**4**
Khepera (Mondada et al., 1994)	IR		
Alice (Caprari and Siegwart, 2003)	IR		
SBot (Mondada et al., 2003)	L		
Jasmine (Kornienko et al., 2005b)	IR		
E-puck (Mondada et al., 2009)	IR, L		**AC**
MarXbot (Bonani et al., 2010)	IR, L		
Kilobot (Rubenstein et al., 2012)	IR		
R-One (McLurkin et al., 2013)	IR		
Droplet (Farrow et al., 2014)		**IR**	
GRITSBot (Pickem et al., 2015)	IR		
Pheeno (Wilson et al., 2016)	AC		
HoverBot (Nemitz et al., 2017)		**MF**	

IR: Infrared Light, AC: Acceleration, L: Visible Light, F: Force, EMF: Electromagnetic Field, MF: Magnetic Field. Other systems that were considered but ended up using "single-measurand sensors": Swarm Bot (McLurkin et al., 2006), Kobot (Turgut et al., 2007), and Thymio-II (Riedo et al., 2013).

In the following section, we introduce an *instrument* model, a well-established model borrowed from the sensors community to generally describe a measuring device, to establish a clear understanding of sensors and how they can become multi-functional. Then we will give a comprehensive review on the sensing capabilities of previous swarm robotic systems and categorize them based on their multi-functionality. Finally, we show an implementation in which a HoverBot (Nemitz et al., 2017) extracts multi-functionality from a single Hall-effect sensor. We show that we can associate time-based magnetic field measurements to robot rotation, collision, and successful movement; and we show how to build a corresponding time-based classifier that can be trained offline before it is transferred to HoverBot for online classification. We apply our method to the HoverBot platform and discuss its applications to time-series data.

1.2. Instrument Model

The *instrument model* shown in **Figure 1** is a scientifically accepted model from the sensor community (Webster, 1999) to generally describe a measuring device. An instrument is a device that transforms a physical variable of interest, the *measurand*, into a form that is suitable for recording, the *measurement*, as conceptually shown in **Figure 1A**. An example of a basic instrument is a ruler. In this case the measurand is the length of some object and the measurement is the number of units (meters, inches, etc.) that represent the length.

Any measurand (distance, collision, temperature, etc.) is linked to an *observable physical measurement variable* X. The observable physical measurement variable X does not necessarily have to be the measurand, X can only be related to the measurand. For instance, the mass of an object is often measured by the process of weighing, where the measurand is the *mass* but the physical measurement variable is the downward *force* the mass exerts in the Earth's gravitational field. Collision detection is another example. A robot can detect the measurand *collision* by measuring *force* or by relating the measurand to another physical measurement variable such as *acceleration*. In this case you can either purchase a single accelerometer or a set of force sensors (e.g., four force sensors – one sensor on each robot side). Both implementations allow the detection of collisions, however, the single accelerometer is likely going to be cheaper than the force sensors. There are many more of such examples, but there are also variants in which a single physical measurement variable contains information about several measurands. An excellent example is IR light. IR light can be used for the measurement of distance, to determine bearing and to communicate with other robots as shown by (Farrow et al., 2014). We call this capability *multi-functional sensing*. Communication is usually handled by a transceiver; you transmit and receive or *transceive* data by means of a physical channel e.g., by utilising electromagnetic waves in the IR spectrum. The *receiving* of signals requires sensors, such as photodiodes that transduce IR light into electric signals, hence communication itself can be considered as a sensing task.

1.3. Sensing Capabilities of Swarm Robotic Systems

We reviewed the sensing capabilities of 15 swarm robotic systems found in the literature and we summarize our findings in Table

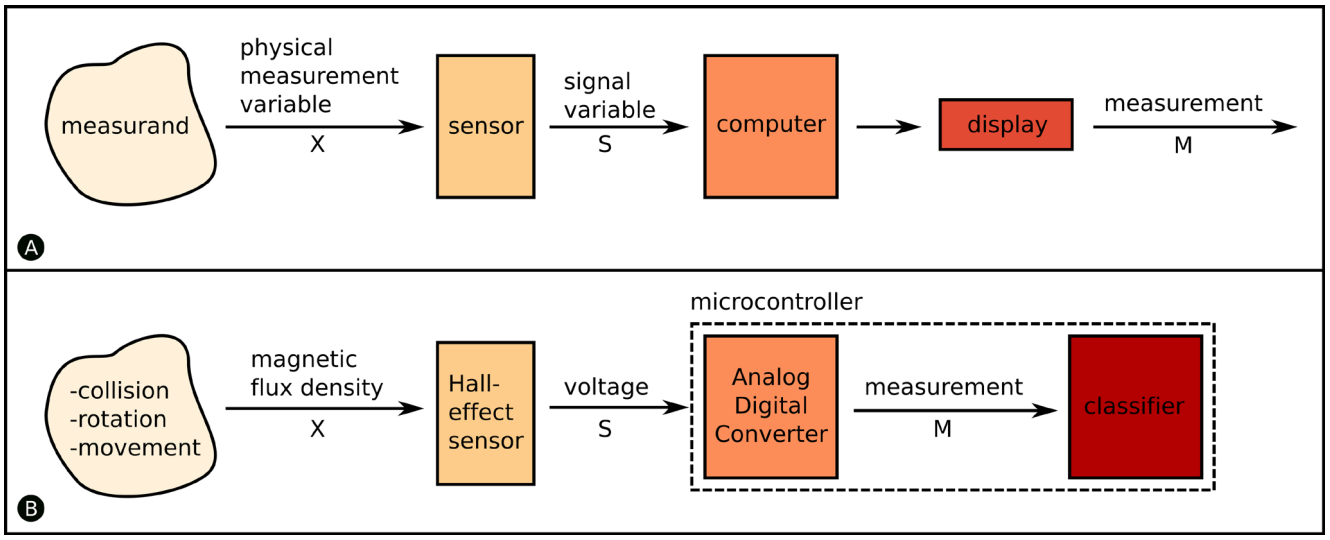


FIGURE 1 | Instrument Model (Webster, 1999). **(A)** The measurand is the to-be-measured value of interest, whereas the physical measurement variable is associated, either directly or indirectly, to the measurand. The sensor converts the physical measurement variable in a signal variable (often an electric signal), and feeds it into a processing unit or computer. The value that we actually display is the measurement. **(B)** Analog to A. We detect collision, rotation and movement measurands by measuring magnetic flux density. The Hall-effect sensor converts the magnetic field into a voltage which is converted through the microcontroller's ADC to a digital signal. We finally apply the measurements to a classifier which associates the measurements to one of the three measurands.

S1 of the supplemental material. **Table 1** is a subset of Table S1 containing robot systems capable of multi-functional sensing with ≥2 measurands. The content of **Table 1** is based on the cited work shown in the first column of each row. **Table 2** takes an even closer look at the few robot systems capable of multi-functional sensing with ≥3 measurands.

The majority of swarm robotic systems (12/15) are capable of multi-functional sensing with ≥2 measurands. In the ≥2 measurand category, the most commonly used *physical measurement variables* are IR light, followed by ambient light. IR light has been mainly utilised for distance/proximity sensing and local communication, whereas ambient light has been often utilised for object detection and long-range distance measurements through a camera.

To the best of our knowledge, there are only two swarm robotic systems that are capable of, or make use of, multi-functional sensing with ≥3 measurands. The e-puck is capable of measuring four measurands with a single sensor (Mondada et al., 2009). It uses a single accelerometer to measure *inclination*, *collision*, *free-fall* and *movement acceleration*. The Droplet is capable of measuring three measurands with a set of IR sensors (Farrow et al., 2014). It uses six symmetrically placed IR sensors to measure *distance*, *bearing* and local *communication*. Therefore, once a robot possesses an accelerometer and a group of IR sensors, it is capable of measuring seven measurands by only using two different types of sensors.

TABLE 2 | Further comparison of swarm robotic systems with 3 or more measurands per sensor.

Measurand / Physical Measurement Variable	IR	AC	MF
	Droplet	E-puck	HoverBot
Local Communication	X		
Proximity/Distance	X		
Bearing	X		
Inclinometer		X	
Collision		X	X
Free-fall		X	
Movement acceleration		X	
Odometry			X
Rotation			X

IR: Infrared Light, AC: Acceleration, MF: Magnetic Field

Our work, which we present here, adds another system to the ≥3 measurand category. **Figure 1B** indicates HoverBot's instrument model. The measurands *successful movement, collision* and *rotation* can be related to the physical measurement variable *magnetic flux density*. HoverBot uses a Hall-effect sensor to convert the magnetic flux densities into voltages which are subsequently converted via an analogue digital converter to digital measurements. The microcontroller processes the samples and checks them against a previously trained classifier. Our classifier combines dynamic time warping and barycenter averaging to build time-variant representations of the measurands. **Figure 2** gives an overview of all swarm robotic systems and their sensing capabilities.

The instrument model comes with its limitations: the *measurand* component can be interpreted from different angles. For example, successful movement and rotation can be generally considered as being part of a robot's *odometry* capability. However, the to-be-measured value, the measurand, is not odometry but successful movement and rotation. Since the instrument model is to some extent subjective, it is of paramount importance to apply this model consistently. **Figure 2** is a collection of carefully categorised sensing capabilities of swarm robotic systems. A key understanding that we have derived from studying the instrument model is that, in swarm robotic systems, often the sensors could be further utilised, and therefore the systems should be reanalysed.

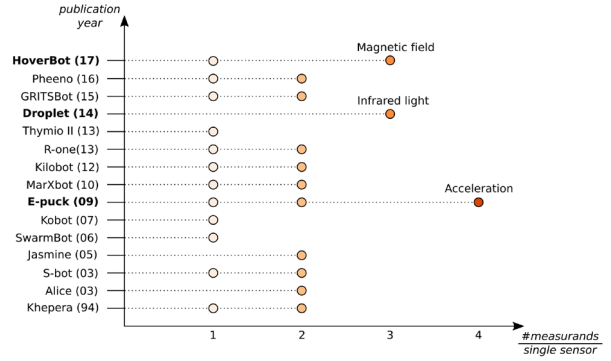


FIGURE 2 | Overview of the sensing capabilities of previous swarm robotic systems. Swarm robots are sequentially listed according to their publication date. The x-axis indicates how many measurands can be measured per single sensor. Although most swarm robotic systems utilise their sensors to detect up to two measurands, there are only three robot systems including the HoverBot that utilise their sensors to measure 3 or more measurands.

1.4. Applying Multifunctional Sensing

Our work empirically shows how HoverBot's single Hall-effect sensor can be augmented for the detection of *collision*, *rotation*, and *successful movement*. Although we specify in the discussion under which circumstances our techniques might be applied to other sensors, we would like to give a brief insight into the research opportunities that potentially arise from (i) HoverBot's new sensing capabilities and (ii) our approach more generally.

1.4.1. Collision Dependent Behaviours

Rotation and *successful movement* detection are proprioceptive sensing capabilities, hence give insight into the internal state of a robot. Counting successful movements is useful for the robot to keep track of its position (odometry), and detecting rotations to derive knowledge about its orientation. Detecting *collisions* is an exteroceptive sensing skill. It provides the robot with information about its surroundings. We give a brief overview over the few swarm robotic studies that deal with collisions and indicate how they were utilised. Kernbach et al. and Schmickl et al. worked on the re-embodiment of biological aggregation behaviours of honeybees. They show how to take advantage of collisions to develop scalable robot behaviours. In their work, swarm robots converge to light sources without requiring inter-robot communication. Concretely, they minimize sensing and computation by evaluating robot data only once per *collision*; more frequent *collisions* lead to more data evaluations (Kernbach et al., 2009; Schmickl et al., 2009). Mayaa et al. harnessed collisions to help localise a robot within an arena. The arena was divided into differently sized segments, whereas each segment was inhabited by differently sized robot groups. Robots used collision detection as information source to determine their locations (Mayya et al., 2017).

Overall, *collision* is a promising candidate for research on and the design of *scalable* robot behaviours since collisions incidences usually increase with increasing group sizes. *Scalable* refers to the ability of a swarm to perform well with different group sizes; the introduction or removal of individuals does not result in drastic change in the performance of a swarm (Brambilla et al., 2013). Collisions have only been sparely studied in the swarm robotics context. Kernbach, Schmickl and Mayaa et al.'s work depict excellent starting points for future work on collisions; the HoverBot system depicts a suitable research platform since it embraces collisions and is capable of detecting them.

1.4.2. Collective Perception

Other interesting work that might profit from our approach is research on collective perception. Collective perception broadly refers to collectives that explore an environment and evaluate its features (Valentini et al., 2016). The work presented here has the potential to enhance a robot's sensing capabilities without modifying its hardware, hence, could add to the list of observable features for collective perception. Notable literature on collective perception includes Khaluf's work on detecting and marking features e.g., of pollution areas (Khaluf, 2017), Kornienko et al.'s work on investigating which sensing and processing steps should be done individually or collectively for collective perception with robot swarms (Kornienko et al., 2005a), Schmickl et al.'s work on hop-count and Trophallaxis-inspired strategies to collectively perceive targets (Schmickl et al., 2007), Mermoud et al.'s work on aggregation-based strategies to collectively perceive and destroy specific targets (Mermoud et al., 2010), and Tarapore et al.'s work on collective perception strategies inspired by the adaptive immune response to discriminate between dangerous and friendly cells (Tarapore et al., 2013).

2. THE HOVERBOT SYSTEM

2.1. Review

We reported in (Nemitz et al., 2017) on the HoverBot system. The HoverBot system is a swarm robotic system and the first of its kind that uses *active low-friction locomotion*. Active low-friction locomotion supplies robots with a constant air flow beneath their surfaces. The airflow causes a reduction of friction between robot and arena surface allowing relatively weak forces to be used for locomotion. In addition, we embedded permanent magnets into the arena as indicated in **Figure 3A**. The HoverBot is a levitating circuit board that possesses planar coils that interact with the arena magnets, resulting in two-dimensional locomotion. Such forces would be insufficient if friction had not been reduced. From the outset, the HoverBot system was designed for manufacturability: HoverBots only require electronics components that are surface mountable, only require connecting a battery to a robot as an assembly step, use low-cost actuators and associated circuitry, do not require actuator calibration and move precisely on a discrete grid. For more details, please refer to our publication (Nemitz et al., 2017).

The work presented here focuses on HoverBot's sensing capabilities. The HoverBot is equipped with IR and Hall-effect sensors as shown in **Figure 3A**. The IR sensor points upwards and therefore only allows communication to an overhead IR handheld that is connected to a PC rather than to other robots. The Hall-effect sensor is positioned in the centre of the HoverBot

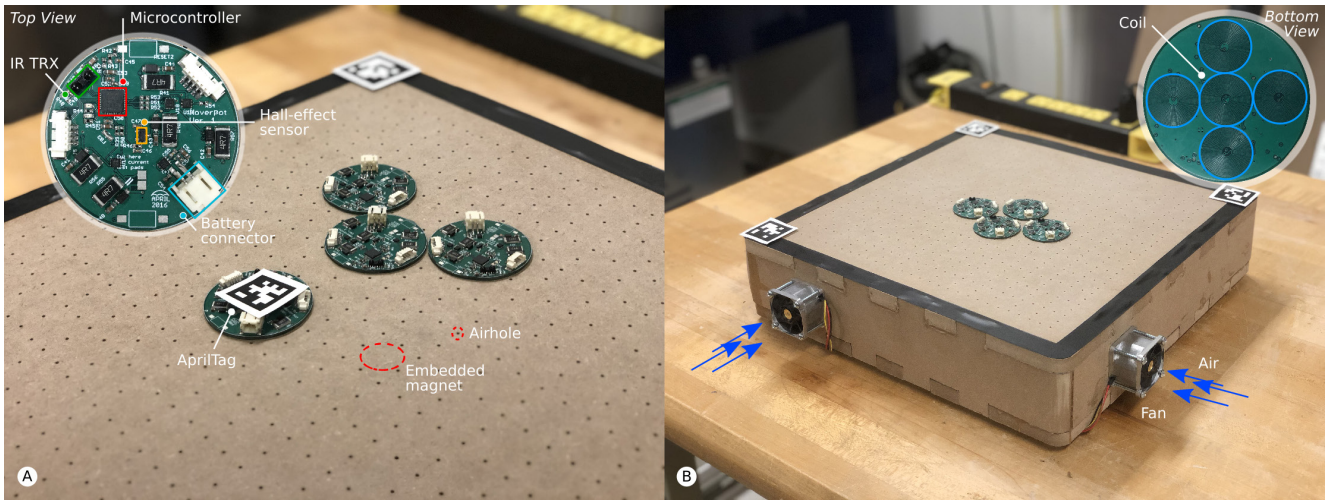


FIGURE 3 | The HoverBot system. **(A)** The HoverBot is displayed in detail in the top left corner. It consists of a low-cost microcontroller, an IR transceiver and a Hall-effect sensor. Permanent magnets are embedded into the platform and air holes are drilled through the surface as exemplary indicated through red circles. We placed AprilTags on a HoverBot as well as in three of the four corners of the magnet-levitation table. This setup allows us to keep track of HoverBot's position during experiments. **(B)** The bottom side of the HoverBot is displayed in the top right corner. A HoverBot possesses five planar coils that it uses to manoeuvre two-dimensionally on the magnet-levitation table. We installed four fans, one on each side of the levitation-magnet table. The fans force air into the magnet-levitation table creating a pressure differential between the inside and outside of the table. Air streams through the porous surface of the magnet-levitation table creating air-cushions beneath HoverBots which makes the robots levitate.

agent and measures the ambient magnetic flux density. Simplified, a Hall-effect sensor is a transducer that converts a magnetic field into a voltage difference, whereas magnetic fields, also called magnetic flux densities, are measured in Tesla [N/m^2] or Gauss [1T = 10,000G].

2.2. The Magnetic Field

HoverBots energize their coils to pull themselves towards magnets. We simulated and subsequently took measurements from the magnetic field. **Figure 4B** depicts the simulated magnetic field acquired with FEMM, a simulator for solving low frequency electromagnetic problems on two-dimensional planar and axisymmetric domains (Meeker, 2010). The grey rectangles in **Figure 4B** indicate magnets and the dashed lines serve as reference to the measurements in **Figure 4C**.

We obtained the magnetic field in **Figure 4C** with a HoverBot. The simulated and measured magnetic fields broadly correlate with one another. The first amplitude in **Figure 4C** is slightly shifted due to fabrication tolerances. The pocket holes of the permanent magnets are slightly larger in diameter than the magnets themselves leading to imperfect magnet alignments. For more details on the manufacture of the HoverBot system please see our previous paper (Nemitz et al., 2017).

2.3. Magnetic Field Profiles

During operation, a HoverBot agent measures magnetic field values in the range of approximately −20 to +60 mT as shown **Figure 4C**. However, the actual measurements during movement look somewhat different since they are not only *position-dependent* as in **Figure 4C** but also *time-dependent*. **Figure 5** shows a set of time-dependent magnetic field measurements. **Figure 5A** indicates the magnetic field measurements of a HoverBot during a successful movement from one permanent magnet to another. When a HoverBot agent collides with an object, its magnetic field measurements look distinctly different compared to its successful movement measurements as shown in **Figure 5B**. We reported in (Nemitz et al., 2017) that HoverBot could potentially lose their

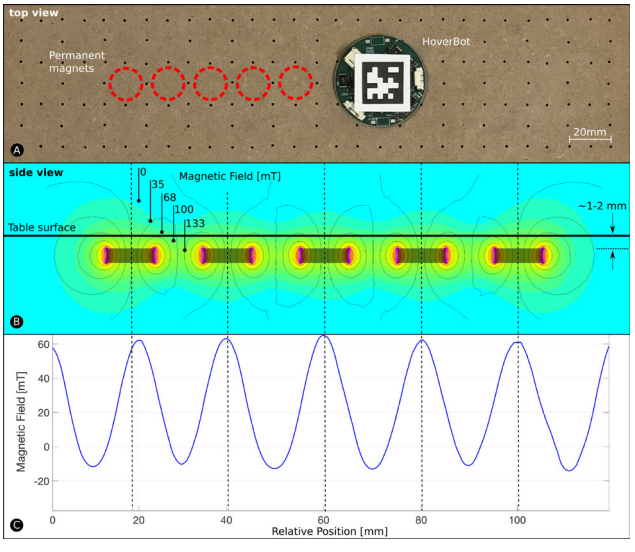


FIGURE 4 | The Magnetic Field. **(A)** HoverBot moves on its magnet-levitation table. **(B)** We simulated HoverBot's magnetic field measurements via finite element method. The permanent magnets are embedded into the magnet-levitation table and are manufactured to be ~1–2 mm beneath the surface. **(C)** We took magnetic field measurements from a HoverBot during movement.

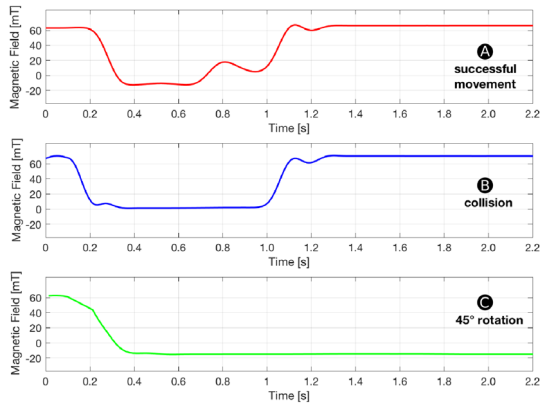


FIGURE 5 | Magnetic Field Profiles. This figure shows examples of magnetic field measurements over time that a HoverBot measures during **(A)** a successful movement **(B)** a collision and **(C)** a 45-degree rotation. Each time series is distinctly different from the other.

orientation, rotate 45 degrees, and lock into position. **Figure 5C** indicates the magnetic field measurements that occur in this incident.

The magnetic field profiles for successful movement and collision in **Figure 5A,B** possess distinct magnetic field changes over time. The magnetic field profile for rotation in **Figure 5C** differs from the other profiles since the magnetic field does not change anymore once the robot rotated and locked into position. Therefore, detecting rotation is a simple case of measuring a constant negative magnetic field over time. Since detecting rotation is trivial and does not add value to this work, we explicitly decided to focus on the remaining more challenging profiles. The remaining work focuses on the classification of time series data for successful movements and collisions.

2.4. Data Acquisition

One of HoverBot's advantages is its precise locomotion in Manhattan Geometry (Nemitz et al., 2017). Missteps or rotations are rare events. During data acquisition, we only observed successful movements and collisions. The robot randomly moved within the arena and occasionally collided with arena boarders. Our dataset consists of 259 samples, whereas 203 samples are from successful movements and 56 samples are from collisions. Each sample is a collection of timestamps, magnetic field measurements, x-position, y-position and orientation of the HoverBot. We acquired our data with the experimental setup in (Nemitz et al., 2017). We placed an artificial fiducial, an AprilTag (Olson, 2011), onto the HoverBot. AprilTags are simplified 2D barcodes that are robust to occlusions and lens distortion allowing high detection rates with camera systems - 20 Hz in our setup. To measure the trajectory of a HoverBot, we tracked the centroid and the orientation of the robot's AprilTag. We used a Chameleon 1.3 MP Color (Sony IXC445) camera and a Tamron 13FM28IR 2.8 mm f/1.2 day/night lens. We also installed an IR transceiver above the arena and connected it to a centralised PC to record HoverBot's magnetic field measurements; HoverBot transmitted during runtime its magnetic field measurements *online* to the overhead IR transceiver. The camera system and the IR transceiver were embedded into LCM (Huang et al., 2010) providing us with a robust data acquisition platform for our experiments.

3. TIME SEQUENCE CLASSIFICATION

A major challenge in discriminating between successful movement and collision profiles is their variations in the time and measurement domain. For example, a HoverBot's speed may vary between actuations, resulting in measurement signals that are stretched or compressed. Additionally, environmental factors can cause measurement values to vary over time, such as when HoverBot's elevation above the table (and hence, its position in the magnetic field) varies due to air pressure fluctuations in the magnet-levitation table. To classify the measurement profiles depicted in **Figure 5A,B**, we must account for these variations while operating within the computational constraints of HoverBot's microcontroller.

We describe in the following sections a classification method that learns offline representations of each measurement profile and stores these representations on the robot. When the HoverBot agent obtains a new series of measurements online, the measurements are compared to the stored class representations to determine the maximum likelihood classification. Our classification procedure builds on several component techniques from the field of signal processing. We introduce each of these components in turn in Section 3.1 and then incorporate them into our method in Sections 3.2 and 3.3. Throughout Section 3.1 we refer to **Figures 6 and 7** for specific examples.

3.1. Classification Preliminaries

3.1.1. Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) (Chiba and Sakoe, 1978) is used to align two time sequences of potentially different length and measure the amount of similarity between them. DTW finds correspondences between points from the two sequences by warping them in the time domain. Given a distance measure, DTW computes the set of point correspondences that minimizes the cumulative distance between the sequences. **Figure 6** shows an example of DTW for two arbitrary signals.

Consider two input signals $\boldsymbol{x} = [x_1, \cdots, x_M]$ and $\boldsymbol{y} = [y_1, \cdots, y_N]$. Let $d(x_i, y_j)$ be a measure of the distance between x_i and y_j, such as the squared Euclidean distance $d(x_i, y_j) = (x_i - y_j)^2$. First, DTW computes the *distance matrix* d (**Figure 6A**) and subsequently the *accumulated distance matrix* D (**Figure 6B**). DTW initialises the first row and column of D as follows:

$$D_{1,1} = d(x_1, y_1)$$
$$D_{i,1} = D_{i-1,1} + d(x_i, y_1), \quad i = 2, \cdots, M$$
$$D_{1,j} = D_{1,j-1} + d(x_1, y_j), \quad j = 2, \cdots, N$$

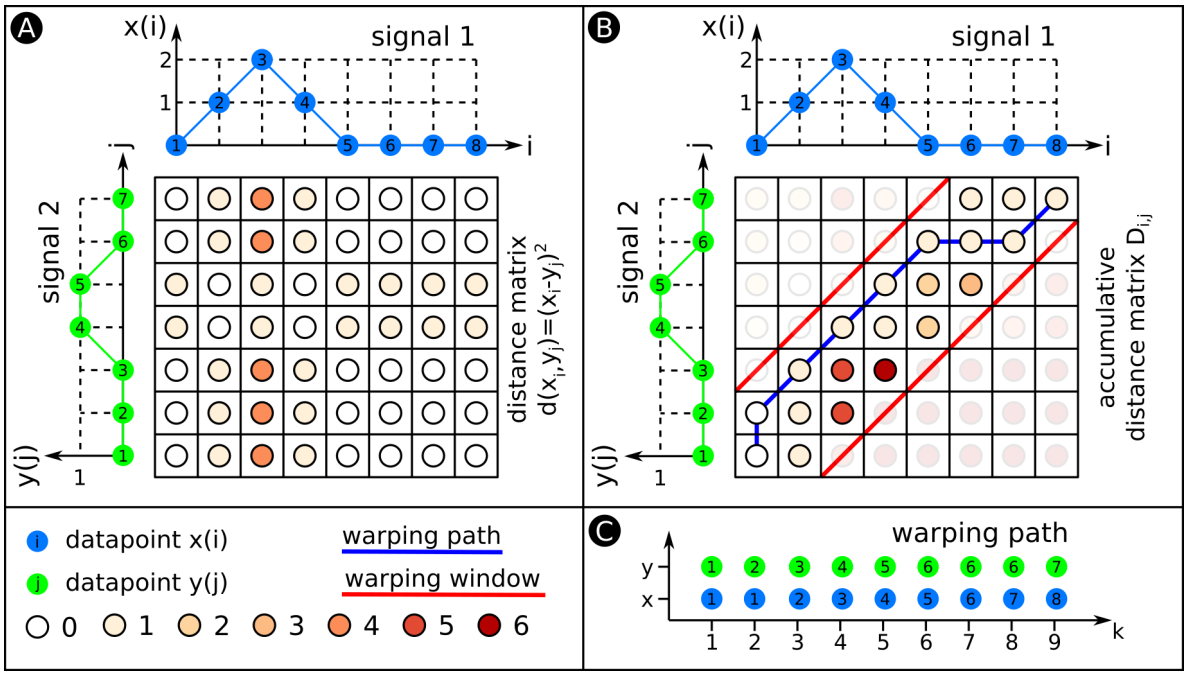


FIGURE 6 | Dynamic Time Warping (DTW). This figure exemplarily explains DTW. **(A)** We compute a matrix in which each entry is the Euclidean distance between datapoints from signals 1 and 2. The colour code for the various distances is shown below. **(B)** We use the distance matrix from **(A)** to develop the accumulated distance matrix D. The two red lines indicate the warping window. The warping path illustrated as blue line has to fit within the borders of the warping window. **(C)** The warping path explicitly states which datapoints of signal 1 align with what datapoints of signal 2. The warping path is always at least as long as the longest signal that is warped. In this example, the warping path exceeds both signals' length, the warping path is 9 datapoints long.

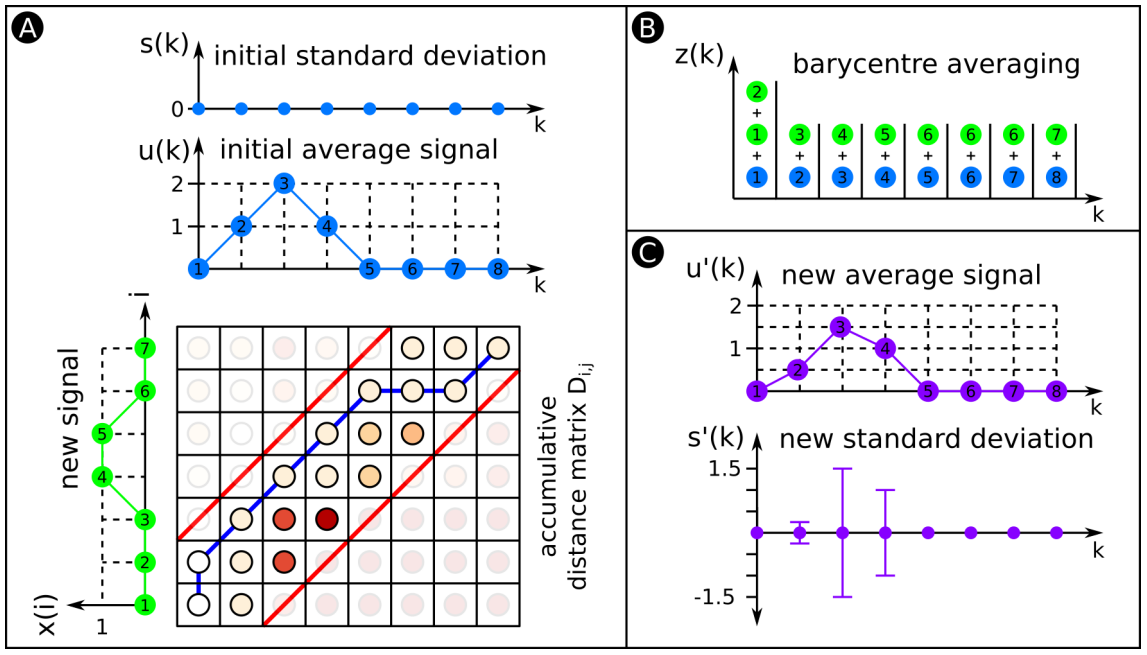


FIGURE 7 | DTW Barycentre Averaging (DBA). **(A)** Similar to **Figure 6B**. This time the signals are initial average signal u(k) and new signal x(i), whereas x(i) will be aligned and averaged with u(k). **(B)** DBA performs DTW to align both signals. The warping path indicates which datapoints from x(i) are aligned with what datapoints from u(k). There is a container for each k; we copy the datapoints that are warped to a specific k (see **Figure 7A**) into the corresponding container. **(C)** We compute from the datapoints of each container the average signal and SD. To build the average signal and SD of a group of signals, warp all signals with the initial average signal. This process fills the various containers with many more datapoints, however, the technique remains the same.

The rest of D is then computed as:

$$\mathrm{D}_{i,j} = d\left(x_i, y_j\right) + \min \begin{cases} D_{i-1,j} \\ D_{i,j-1} \\ D_{i-1,j-1} \end{cases}, \quad \begin{matrix} i = 2, \cdots, M \\ j = 2, \cdots, N \end{matrix}$$

The DTW algorithm uses dynamic programming to build D, where each element

$D_{i,j}$ gives the minimum cumulative distance between subsequences $\left[x_1, \cdots, x_i\right]$ and $\left[y_1, \cdots, y_j\right]$. The final element of D, $D_{m,n}$, represents the minimum cumulative distance between the signals under any warping configuration, thus serving as a measure of similarity between the signals. Such an optimal warping configuration is given by the warping path $\boldsymbol{\phi}^{x,y} = \left[\left(\phi_1^x, \phi_1^y\right), \cdots, \left(\phi_K^x, \phi_K^y\right)\right]$ through matrix D (**Figure 6C**). Each element $\phi_k^{x,y} = \left(\phi_k^x, \phi_k^y\right)$ contains an index ϕ_k^x of signal x that has been matched to index ϕ_k^y of signal y. The warping path can be computed as:

$$\phi_K^{x,y} = \left(M, N\right)$$

$$\phi_k^{x,y} = \underset{i,j}{\operatorname{argmin}}\, D_{i,j} \;\; i,j = \begin{cases} \phi_{k+1}^x - 1, \phi_{k+1}^y \\ \phi_{k+1}^x, \phi_{k+1}^y - 1 \\ \phi_{k+1}^x - 1, \phi_{k+1}^y - 1 \end{cases}, \; k = K-1, \cdots, 2$$

$$\phi_1^{x,y} = \left(1, 1\right)$$

The length K of warping path $\boldsymbol{\phi}^{x,y}$ varies depending on the extent to which the signals are warped, but it will satisfy the inequality $\max\left(M, N\right) \leq K \leq M + N - 1$.

3.1.2. Constrained Dynamic Time Warping (CDTW)

The DTW formulation of the previous section is *unconstrained* i.e. the algorithm considers any possible warping configuration in matrix D. This can lead to *pathological* warping configurations wherein a single point from one signal is matched with many points from the other signal. Unconstrained DTW is thus sensitive to spurious alignments between signals. For example, unconstrained DTW might warp the collision time series in **Figure 5B** onto the successful movement time series in **Figure 5A** even though the signals correspond to disparate events. Given these limitations of unconstrained DTW, it is common to limit the extent to which signals can be warped in the time domain. In Constrained Dynamic Time Warping (CDTW), we allow correspondences between points only if those points occur within a fixed time period of one another. The length of the warping window W is application-dependent and defined as $|n - m| \leq W$. W determines how many elements of matrices d (distance matrix) and D (accumulated distance matrix) are calculated. We show an example of a warping window in Figure 6b. CDTW can offer benefits in terms of time- and space-complexity, both are important for embedded platforms like the HoverBot. Because the warping path is constrained, only a portion of the matrix D must be computed and stored (Chiba and Sakoe, 1978). As a result, the time- and space-complexity are both reduced from $O\left(N \cdot M\right)$ to $O\left(N \cdot W\right)$, where the length of the warping window W is much less than M, the length of signal $\boldsymbol{y}$.

3.1.3. DTW Barycenter Averaging (DBA)

For a classification task, it is often useful to compute a summary representation of a class of data. This averaging process is non-trivial when performed on variable-length signals. Naïve approaches based on pairwise alignment and averaging are sensitive to ordering effects and produce prohibitively long alignment sequences (Petitjean et al., 2011). We instead utilize DTW Barycenter Averaging (DBA) to compute the average signal $\boldsymbol{\mu}$ and the standard deviation $\boldsymbol{\sigma}$ of a group of signals (Morel et al., 2018). **Figure 7** shows an example of DBA for *two* arbitrary signals.

Consider a group of signals X to be input to the DBA algorithm. DBA initializes $\boldsymbol{\sigma}$ to be zero and randomly selects a signal $\boldsymbol{x}^{(0)} = \left[x_1^{(0)}, \cdots, x_K^{(0)}\right]$ to serve as the initial average signal $\boldsymbol{\mu}$ (**Figure 7A**) as

$$\mu_k = x_k^{(0)}, \; k = 1, \cdots, K$$

DBA also initializes sets $\psi_k = \varnothing, k = 1, \cdots, K$ that are used in the average computation. Each set ψ_k contains $z \in \psi_k$ elements for each specific $k \in K$, whereas z is a placeholder for aligned datapoints (**Figure 7B**). Each signal $\boldsymbol{x} \in X \backslash \boldsymbol{x}^{(0)}$ is aligned with the average signal using CDTW. Each set ψ_k is updated as

$$\psi_k = \psi_k \cup \left\{ x_{\phi_i^x} : \; \phi_i^\mu = k \right\}$$

Please note that our example in **Figure 7** only shows DBA for *two* signals. In **Figure 7B**, set ψ_1 consists of three aligned datapoints and sets ψ_{2-8} of two aligned datapoints. Once this process has been repeated for *all* signals, the value of each element of the average signal is updated as the barycenter of all points that map to the corresponding element of the existing average signal $\boldsymbol{\mu}$ (**Figure 7C**); that is,

$$\mu_k = \frac{\sum_{z \in \psi_k} z}{\left|\psi_k\right|}$$

Likewise, the elements of the standard deviation $\boldsymbol{\sigma}$ (**Figure 7C**) are updated as

$$\sigma_k = \sqrt{\frac{\sum_{z \in \psi_k} \left(z - \mu_k\right)^2}{\left|\psi_k\right|}}$$

The resulting updated average signal retains its original length while incorporating information from the entire group of signals. The process of aligning the group of signals with the average signal and computing an updated average and SD can be repeated multiple times for better convergence.

3.1.4. Downsampling

Our method can be implemented in several variants. Unconstrained with a complexity of $O\left(N \cdot M\right)$, constrained with a complexity of $O\left(N \cdot W\right)$ using a warping window, or even further constrained with a complexity of $O\left(K \cdot W\right)$ through a combination of warping window and downsampling. Consider an input signal $\boldsymbol{x} = \left[x_1, \ldots, x_N\right]$ with length N. We reduce the signal's length from N to K by dividing the signal $\boldsymbol{x}$ into K parts, each of which contains $L = \frac{N}{K}$ measurements. The value of each element $x_k^{'}$ of the downsampled signal $\boldsymbol{x}^{'}$ is given by

$$x_k^{'} = \frac{1}{L} \sum_{i=L\left(k-1\right)}^{Lk} x_i\,, \quad k = 1, \ldots, K$$

Downsampling has an impact on the classifier's detection rate. The intuition is, the more you downsample the input signals, the worse the detection rate becomes. Downsampling potentially averages out important signal features that otherwise helped the classifier to discriminate between signals. From an engineering perspective, the more you downsample the input signals, the less memory is required to store values of the cumulative distance matrix. Memory reductions are very useful since it lowers the requirements for our low-cost microcontroller. Hence, there is a tradeoff between classification performance and memory utilization. We are able to achieve high classification performance with acceptable memory usage as shown in the Result section.

3.2. Offline Learning of Class Representations

Figure 8A gives an overview of the components that are involved in the offline learning of class representations. First, we conduct a random-walk HoverBot experiment and record the measurements (x-, y-position, orientation, magnetic field measurement, timestamp). We separate the data into approximately two second intervals, HoverBot's coil actuation scheme, to obtain a dataset consisting of many training examples. We manually label these examples based on HoverBot's movement over time. We also downsample ($K = 20$) each example to accord with the computational limitations of HoverBot's microcontroller.

Given this set of labelled training examples, we perform k-fold cross-validation ($KV = 10$). In each iteration, we divide (partition) the training data into labelled classes and perform DBA to obtain a representation of each class. In the validation step, we compute the Mahalanobis distance between each held-out example and the representations of each class:

$$d\left(\mu_k, \sigma_k, x_k\right) = \sqrt{\sum_{k=1}^{K} \frac{\left(x_k - \mu_k\right)^2}{\sigma_k}}$$

The Mahalanobis distance measures the distance between a point and a distribution (De Maesschalck et al., 2000). In other words, how many standard deviations σ is a point x away from the mean value μ. We classify each example according to the minimum-distance class, which corresponds to the maximum-likelihood classification.

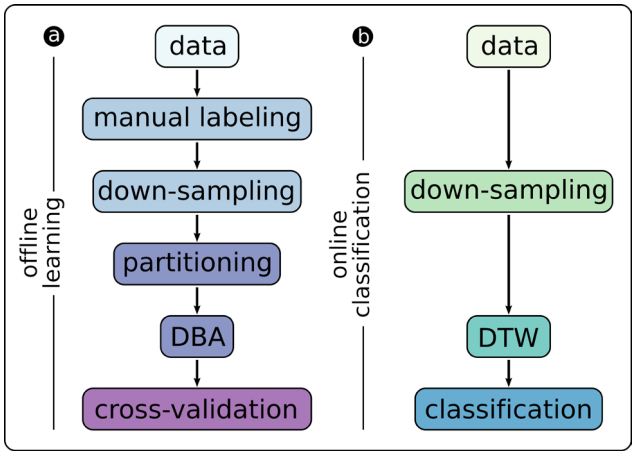


FIGURE 8 | Overview of signal processing components for **(A)** offline training of the classifier and **(B)** online classification of new data samples.

3.3. Online Classification of Hall-Effect Measurements

Figure 8B indicates the components that are involved in the online classification. The components in **Figure 8B** are a subset of **Figure 8A**. HoverBot tries to move into a direction and records magnetic field measurements. Depending on the downsampling frequency, the HoverBot stores a number of average values into its memory (K = 20). The new dataset is used to calculate the Mahalanobis distance for each class representation (successful movement and collision). The Mahalanobis distances are compared with one another. If the distances do not reach a minimum value, the event will be labelled as *unknown*. Otherwise the event will be classified according to the lower Mahalanobis distance. The parameters of the class representations (μ_k, σ_k) were trained offline, do not change during runtime and therefore are stored on the microcontroller's read-only-memory (ROM). The amount of ROM memory involved is dependent on the downsampling frequency.

4. RESULTS

Figure 9 shows the trained class representations for successful movement and collision. For each datapoint our classification method produces a mean value and a SD. These values are constant and are stored in the microcontroller's flash memory.

Figure 10 indicates the detection rates of our successful movement and collision classifier. For each number of datapoints (K) we compute a confusion matrix through k-fold cross-validation as described in the *Offline Learning of Class Representations* Section. The detection rate for a successful movement is calculated by the True-Positive-Rate and the detection rate for a collision by the False-Positive-Rate of the confusion matrix. We give a confusion matrix example for K = 20 in **Figure 10**. While the detection rate increases with the number of datapoints, it starts stagnating once it exceeds 20 datapoints per sample. Therefore, we chose K = 20 in

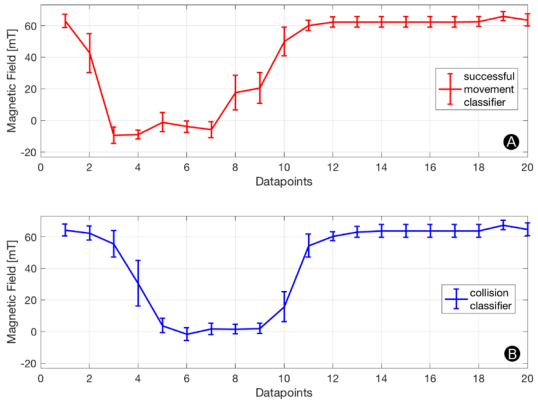


FIGURE 9 | Classifier Parameters. Each datapoint has a mean value and SD associated to it. These constants are stored in the microcontroller's ROM and used for online classification. The graphs indicate how much tolerance (SD) is permitted at each datapoint.

our setup achieving collision and successful movement detection rates of greater 85%. This setup is a fair trade-off between detection rate and computational effort. Our microcontroller has to store 20·20·2 Bytes, assuming a 16-bit unsigned integer for each magnetic field measurement and not applying a warping window which otherwise decreased the memory requirement to 20·W·2 Bytes. HoverBot's microcontroller (Atmel SAMD21E16) comes with 8 kB SRAM which comfortably meets the memory requirements of our classification method.

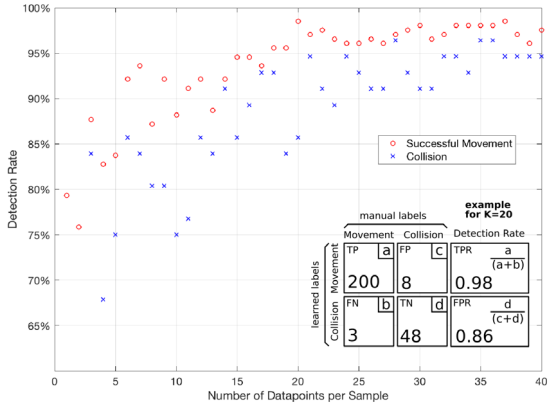


FIGURE 10 | Downsampling. This graph shows the effect of downsampling on the classifier's detection rate. The detection rate exponentially increases until it starts stagnating once it exceeds 20 datapoints per sample. In the bottom right corner of this figure, we show an example of the confusion matrix for 20 datapoints. Confusion matrix legend: TP: True Positive, FN: False Negative, FP: False Positive, TN: True Negative, TPR: True Positive Ratio, FPR:False Positive Ratio.

5. DISCUSSION

This work combines two rather unrelated research fields, the fields of signal processing and swarm robotics. The signal processing techniques were specifically adapted to operate on low-cost hardware. We introduced a warping window and a downsampling method to reduce the classifier's time- and space-complexity. We discuss the impact of downsampling on time varying signals in Section 5.1. We hope that our work encourages other swarm roboticists to reanalyse the sensing capabilities of their robots. The requirements and limitations of our approach are discussed in Sections 5.2 and 5.3. There might be an advantage of adding dedicated general purpose integrated circuits (IC) or Field Programmable Gate Arrays (FPGAs) to robot designs to process sensor data on the side to augment a robot's sensing capabilities. The advantages and disadvantages of specialized sensory is discussed in Section 5.4.

5.1. Downsampling

Our downsampling method decreases a signal's length to a number of averaged datapoints. HoverBot's magnetic field profiles for successful movement and collision are simple. The profiles do not contain high frequency features, hence downsampling only had a limited impact on our classifier's detection rate. Classifiers built upon more complicated time-variant data are expected to be more heavily influenced by our downsampling method.

5.2. Applicability to Other Systems

This study has demonstrated how time sequence classification can be used to measure several measurands with a single sensor. Our method's applicability is dependent on *signatures*, unique measurement profiles that can be associated to specific measurands. Although we have not applied time sequence classification to other sensors or robots, we argue that it is generally applicable if the signature (i) contains time varying measurements (ii) is systematically reoccurring and (iii) is distinctly different to other signatures.

Please find an analysis of suitable sensors, a generalised concept about signatures, and hypothetical examples of our approach in the Appendix.

5.3. Limitations

HoverBot's *discrete movement* helped the discovery that signatures can be used to measure several measurands with a single sensor. If HoverBot's movement was *continuous*, we could still chop the measurements into the measurement profiles shown in **Figure 5**, since any continuous movement can be regarded as a finite number of discrete movements with an infinitely small time difference in between them. However, our signatures are constrained to HoverBot's *movement behaviour*. HoverBot moves on a grid in Manhattan geometry which ensures that it measures a reoccurring magnetic field pattern. Other *movement geometries* such as continuous movements in arbitrary directions would impact our signatures and hence the time sequence classification.

5.4. Multifunctional Sensing vs Specialised Sensory

Our *signatures* allow the *binary* detection of successful movements, collisions, and rotations. These measurands can

also be measured with specialised sensors. For example, you could detect collisions through tactile sensors that output *continuous* measurements. However, this is not a limitation of the time sequence classification but of our signatures. If you can find *distinct* signatures for each collision level, the time sequence classification will enable you to discriminate between them. The advantage of specialised sensory however is that it takes away the processing behind the time sequence classification and enables continuous monitoring. For example, our approach first obtains a new data set and subsequently analyses it for collisions; we can only detect collisions once every movement cycle, whereas tactile sensors could detect collisions at any given time. A movement cycle is defined as a discrete movement from one permanent magnet to another. This might have an impact on the reactivity of robots. The disadvantage of specialised sensory is its component cost and corresponding electronics.

6. CONCLUSION

In this study, we analysed 15 swarm robotic systems for their sensing capabilities using the instrument model from the sensing and measurement community. We exemplarily show how the measurements from HoverBot's single Hall-effect sensor can be associated to successful movement, rotation and collision events. We constrain dynamic time warping (DTW) and DTW Barycenter Averaging (DBA) to perform time-series classification on a low-cost microcontroller. These signal processing techniques are generally applicable to time-variant data, however, must be applied to time varying, distinct, and systematically reoccurring measurements to augment a robot's sensing capabilities. We train a classifier offline, transfer its parameters to HoverBot for online classification, and achieve high detection rates. This work shines light on how swarm roboticists can augment sensors by applying computationally constrained signal processing techniques to gain multi-functional sensing capabilities.

AUTHOR CONTRIBUTIONS

MN created the system and is lead author of the work. RM worked on the development of the classifier and helped writing section 3. MS worked on the introduction literature and the HoverBot system. GF worked on the development of the classifier. AH advised generally on the work. EO advised on developing the classifier. AS was the lead advisor and primary editor of the manuscript.

FUNDING

Markus Nemitz gratefully acknowledges support from the Centre in Doctoral Training in Intelligent Sensing and Measurement (EP/L016753/1) UK, Offshore Robotics for Certification of Assets (ORCA) (EP/R026173/1) UK, and the Office of Naval Research (N00014-13-1-0217), USA. Ryan Marcotte is supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 1256260.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/frobt.2018.00055/full#supplementary-material

REFERENCES

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems. Handbook of Nature-Inspired and Innovative Computing*. UK: Oxford University Press.

Bonani, M., Longchamp, V., Magnenat, S., Rétornaz, P., Burnier, D., and Roulet, G. (2010). "The marXbot, a Miniature Mobile Robot Opening New Perspectives for the Collective-Robotic Research," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference*. Taipei, Taiwan: IEEE, 4187–4193.

Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm. Intell.* 7 (1), 1–41. doi: 10.1007/s11721-012-0075-2

Caprari, G., and Siegwart, R. (2003). "Design and Control of the Mobile Micro Robot Alice," in *Proceedings of the 2nd International Symposium on Autonomous Minirobots for Research and Edutainment AMiRE 2003*. Australia: Brisbane, 23–32.

Chiba, S., and Sakoe, H. (1978). "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*. IEEE 43–49.

De Maesschalck, R., Jouan-Rimbaud, D., and Massart, D. L. (2000). The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems* 50 (1), 1–18. doi: 10.1016/S0169-7439(99)00047-7

Farrow, N., Klingner, J., Reishus, D., and Correll, N. (2014). "Miniature Six-Channel Range and Bearing System: Algorithm, Analysis and Experimental Validation" *Robotics and Automation (ICRA), 2014 IEEE International Conference* (IEEE), 6180–5.

Huang, AS., Olson, E., and Moore, DC. (2010). "LCM: Lightweight Communications and Marshalling" *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS* (IEEE), 4057–4062.

Kernbach, S., Thenius, R., Kernbach, O., and Schmickl, T. (2009). Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system. *Adapt. Behav.* 17 (3), 237–259. doi: 10.1177/1059712309104966

Khaluf, Y. (2017). "Edge Detection in Static and Dynamic Environments Using Robot Swarms," in *2017 IEEE 11th International Conference on Self-Adaptive and Self-Organizing Systems*. SASO 81–90.

Kornienko, S., Kornienko, O., Constantinescu, C., Pradier, M., and Levi, P. (2005a). "Cognitive Micro-Agents: Individual and Collective Perception in Microrobotic Swarm," in *IJCAI-05 Workshop on Agents in Real-Time and Dynamic Environments*. Edinburgh, UK, 33–42. http://www.cybertronica.co/sites/default/files/publications/IJCAI-workshop-ARTDE.pdf.

Kornienko, S., Kornienko, O., and Levi, P. (2005b). "Minimalistic Approach towards Communication and Perception in Microrobotic Swarms," in *Intelligent Robots and Systems, 2005*. IEEE 2228–2234.

Mayya, S., Pierpaoli, P., Nair, G., and Egerstedt, M. (2017). "Collisions as Information Sources in Densely Packed Multi-Robot Systems Under Mean-Field Approximations". *Robotics Science and Systems Conference*.

McLurkin, J., Lynch, A. J., Rixner, S., Barr, T. W., Chou, A., Foster, K., et al. (2013). *A Low-Cost Multi-Robot System for Research, Teaching, and Outreach. Distributed Autonomous Robotic Systems*. Berlin, Heidelberg: Springer.

McLurkin, J., Smith, J., Frankel, J., Sotkowitz, D., Blau, D., and Schmidt, B. (2006). "*Speaking Swarmish: Human-Robot Interface Design for Large Swarms of Autonomous Mobile Robots*". *AAAI Spring Symposium* 72–75.

Meeker, D. (2010). *Finite Element Method Magnetics*. Available at: http://www.femm.info

Mermoud, G., Matthey, L., Evans, W., and Martinoli, A. (2010). "Aggregation-mediated collective perception and action in a group of miniature robots," in *International Conference on Autonomous Agents and Multiagent Systems* 599–606.

Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., and Klaptocz, A. (2009). "The E-Puck, a Robot Designed for Education in Engineering," in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions* 59–65.

Mondada, F., Franzi, E., and Ienne, P. (1994). *Mobile Robot Miniaturisation : A Tool for Investigation in Control Algorithms*, eds T. Yoshikawa, and F. Miyazaki (Experimental Robotics III. Springer).

Mondada, F., Guignard, A., Bonani, M., Baer, D., Lauria, M., and Floreano, D. (2003). "SWARM-BOT: From Concept to Implementation" *2003 IEEE/RSJ International Conference* (Las Vegas, NV, USA), 1626–1631.

Morel, M., Achard, C., Kulpa, R., and Dubuisson, S. (2018). Time-series averaging using constrained dynamic time warping with tolerance. *Pattern Recognit.* 74, 77–89. doi: 10.1016/j.patcog.2017.08.015

Nemitz, M. P., Sayed, M. E., Mamish, J., Ferrer, G., Teng, L., McKenzie, R. M., et al. (2017). HoverBots: Precise Locomotion Using Robots That Are Designed for Manufacturability. *Front. Robot. AI* 4. doi: 10.3389/frobt.2017.00055

Olson, E. (2011). "AprilTag: A Robust and Flexible Visual Fiducial System," in *Proceedings - IEEE International Conference on Robotics and Automation* 3400–3407.

Petitjean, F., Ketterlin, A., and Gançarski, P. (2011). A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognit.* 44 (3), 678–693. doi: 10.1016/j.patcog.2010.09.013

Pickem, D., Lee, M., and Egerstedt, M. (2015). "The GRITSBot in Its Natural Habitat - A Multi-Robot Testbed," in *Proceedings - IEEE International Conference on Robotics and Automation* 4062–4067.

Riedo, F., Chevalier, M., Magnenat, S., and Mondada, F. (2013). "Thymio II, a Robot That Grows Wiser with Children," in *Advanced Robotics and Its Social Impacts (ARSO), 2013 IEEE Workshop*. IEEE 187–193.

Rubenstein, M., Ahler, C., Hoff, N., Cabrera, A., and Nagpal, R. (2014). Kilobot: A low cost robot with scalable operations designed for collective behaviors. *Rob. Auton. Syst.* 62 (7), 966–975. doi: 10.1016/j.robot.2013.08.006

Rubenstein, M., Ahler, C., and Nagpal, R. (2012). "Kilobot: A Low Cost Scalable Robot System for Collective Behaviors," in *Proceedings - IEEE International Conference on Robotics and Automation* 3293–3298.

Schmickl, T., Möslinger, C., and Crailsheim, K. (2007). Collective Perception in a Robot Swarm. *Swarm Robotics* 4433, 144–157.

Schmickl, T., Thenius, R., Moeslinger, C., Radspieler, G., Kernbach, S., Szymanski, M., et al. (2009). Get in touch: cooperative decision making based on robot-to-robot collisions. *Auton. Agent. Multi. Agent. Syst.* 18 (1), 133–155. doi: 10.1007/s10458-008-9058-5

Senanayake, M., Senthooran, I., Barca, J. C., Chung, H., Kamruzzaman, J., and Murshed, M. (2016). Search and tracking algorithms for swarms of robots: A survey. *Rob. Auton. Syst.* 75, 422–434. doi: 10.1016/j.robot.2015.08.010

Tarapore, D., Christensen, A. L., Lima, P. U., and Carneiro, J. (2013). "Abnormality detection in multiagent systems inspired by the adaptive immune system," in *AAMAS '13 Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 23–30.

Turgut, AE., Goekce, F., Celikkanat, H., Bayindir, L., and Sahin, E. (2007). "Kobot: A Mobile Robot Designed Specifically for Swarm Robotics Research" *METU-CENG-TR-2007-05* (Ankara, Turkey),

Valentini, G., Brambilla, D., Hamann, H., and Dorigo, M. (2016). Collective Perception of Environmental Features in a Robot Swarm. *International Conference on Swarm Intelligence* 9882, 65–76.

Webster, J. G. (1999). "The Measurement, Instrumentation and Sensors Handbook," in *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 104, Edn. 1, eds G. John, and (Webster, CRC Press and IEEE Press).

Wilson, S., Gameros, R., Sheely, M., Lin, M., Dover, K., Gevorkyan, R., et al. (2016). Pheeno, A Versatile Swarm Robotic Research and Education Platform. *IEEE Robot. Autom. Lett.* 1 (2), 884–891. doi: 10.1109/LRA.2016.2524987

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers

Supplemental Information

Table S1 Caption: Comparison of 15 swarm robotic systems' sensing capabilities

Year	Robot Name	Measurand	Physical Measurement Variable	Sensor
(Mondada, Franzi, and Ienne 1994)	**Khepera**	Proximity/Distance Ambient IR Light Intensity	Infrared Light	IR Sensor
		Odometry	Magnetic Field	Wheel Encoder
(Caprari and Siegwart 2003)	**Alice**	Proximity/Distance Local Communication	Infrared Light	IR Sensor
(Mondada et al. 2003)	**S-bot**	Proximity/Distance	Infrared Light	IR Sensor
		Humidity	Humidity	Humidity Sensor
		Strain	Force	Tactile Sensor
		Torque	Force	Torque Sensor
		Local Communication	Sound	Microphones
		Temperature	Temperature	Temperature Sensor
		Ambient Light Intensity	Visible Light	Light Sensor
		Robot Detection Color Detection	Visible Light	Camera
		Movement Acceleration	Acceleration	Accelerometer
		Tilt	Acceleration	Inclinometer
		Odometry	Magnetic Field	Wheel Encoder
		Global Communication	E. Magnetic Field	Radio
(Kornienko, Kornienko, and Levi 2005)	**Jasmine**	Proximity/Distance Local Communication	Infrared Light	IR Sensor
(McLurkin et al. 2006)	**SwarmBot**	Global Communication	E. Magnetic Field	Radio
		Collision	Force	Bump skirt
		Ambient Light Intensity	Visible Light	Light Sensor
		Vision	Visible Light	Camera
		Local Communication	Infrared Light	IR sensor

(Turgut et al. 2007)	**Kobot**	Global Communication Local Communication	E. Magnetic Field	Antenna (XBee)
		Proximity	Infrared Light	IR Sensor
(Mondada et al. 2009)	**E-puck**	Proximity Ambient IR Light Intensity	Infrared Light	IR Sensor
		Inclination Collision Detection Free-Fall Detection Movement Acceleration	Acceleration	Accelerometer
		Vision Long Range Distance	Visible light	Camera
		Localisation	Sound	Microphones
(Bonani et al. 2010)	**MarXbot**	Proximity/ Distance Bearing	Infrared Light	IR sensor
		Odometry	Acceleration	Accelerometer + Gyroscope
		Strain	Force	Force sensor
		Vision Distance	Visible Light	2xCameras
		Global Communication	E. Magnetic Field	Bluetooth, WiFi
		RFID Tag Information	E. Magnetic Field	RFID Reader
(Rubenstein, Ahler, and Nagpal 2012)	**Kilobot**	Local Communication Distance	Infrared Light	IR Sensor
		Ambient Light	Visible Light	Light Sensor
(McLurkin et al. 2013)	**R-one**	Local Communication Localisation	Infrared Light	IR Sensor
		Odometry	Acceleration	Accelerometer + Gyroscope
		Robot Position	Visible Light	Camera
		Ambient Light	Visible Light	Light Sensor
		Odometry	Visible Light	Wheel Encoder
		Global Communication	E. Magnetic field	Radio
	Thymio II	Proximity	Infrared Light	IR Sensor

(Riedo et al. 2013)		Global Communication	Infrared Light	IR Receiver (remote control)
		Acceleration	Acceleration	Accelerometer
		Touch	Force	Capacitive Touch Sensor
		Temperature	Temperature	Temperature Sensor
		Sound	Sound	Microphone
(Farrow et al. 2014)	**Droplet**	Distance Bearing Local Communication	Infrared Light	IR Sensor
(Pickem, Lee, and Egerstedt 2015)	**GRITSBot**	Distance Bearing	Infrared Light	IR Sensor
		Odometry	Acceleration	Accelerometer + Gyroscope
		Battery Level	Voltage	Battery Voltage Sensor
(Wilson et al. 2016)	**Pheeno**	Odometry	Magnetic Field	Wheel Encoder
		Heading	Magnetic Field	Magnetometer
		Movement Acceleration Odometry	Acceleration	Accelerometer
		Proximity	Infrared Light	IR Sensor
		Object Identification	Visible Light	Camera
		Global Communication	E. Magnetic Field	WiFi
(Nemitz et al. 2017)	**HoverBot**	Global Communication	Infrared Light	IR sensor
		Odometry Collision Detection Rotation Detection	Magnetic Field	Hall-effect Sensor

Appendix G

Conference: HoverBots: Embracing and Detecting Collisions Using Robots Designed for Manufacturability

Conference: International Conference on Robotics and Automation (2018)

Workshop: Swarms: From Biology to Robotics and Back

Dr. Justin Werfel *reviewer*
Senior Research Scientist
Wyss Institute
Harvard University
United States

Prof. Kirstin Petersen *reviewer*
Assistant Professor
Department of Electrical and Computer Engineering
Cornell University
United States

Prof. Mike Rubenstein *reviewer*
Assistant Professor
Electrical Engineering and Computer Science
Northwestern University
United States

HoverBots: Embracing and Detecting Collisions Using Robots Designed for Manufacturability

Markus P. Nemitz[1], Edwin Olson[2], and Adam A. Stokes[3]

***Abstract*— Collisions play a crucial role in nature. While some natural systems utilise collisions to achieve collective behaviours such as cell migration, most robot systems avoid them. There have been a few studies on collisions with swarm robots. Robot behaviours were collision dependent, however, physical collisions were still avoided. Robots detected close field objects with proximity sensors and accounted them for collisions. However, true collisions cause physical interactions amongst robots and their immediate environment; collision chains might even displace many robots at the time and possibly change the outcome of an experiment; approximating collisions neglects their physical impact on the real world. In this work, we introduce the HoverBot system. HoverBots are floating circuit boards capable of autonomous movement by energising their planar coils to interact with permanent magnets that are embedded into the arena surface. HoverBots embrace physical interactions with other robots or objects. We show how HoverBots utilise magnetic field readings from a Hall-effect sensor to detect collisions and briefly discuss how collisions could be used to map environments.**

I. Introduction

A. Collisions in Biological Systems

There are several examples in nature, where collisions occur amongst biological agents. For example, ants physically interact with one another while building streets or proceeding to raids, fish mildly collide during rapid schooling manoeuvres [1], and people collide while navigating through crowds [2]. Some studies indicate a major influence of collisions on collective behaviours. For example, research on cell migration suggests that cell migration itself is an emergent behaviour, whereas it is evoked by inelastic collisions between neighbouring cells [3]. Collective migration of eukaryotic cells plays a fundamental role in tissue growth, wound healing and immune response. A study on granular media makes comparisons to biologically inspired interacting agents and shows that simple inelastic collisions between self-propelled agents can provide a wide range of self-organised collective behaviours [4].

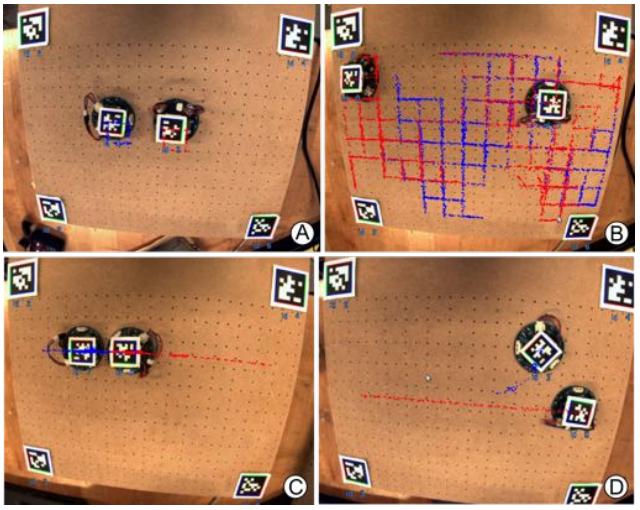


Fig. 1. Demonstration. Red and blue trajectories depict HoverBot's movements over time. (A) Two HoverBots circle in formation until they are unsynchronized; (B) two HoverBots move randomly and collide; (C) two HoverBots collide frontally with one another; (D) one HoverBot collides with a passive HoverBot.

B. Collisions in Swarm Robotics

While collisions naturally occur in nature, most robot systems avoid collisions to keep the robot and its immediate environment safe; collision avoidance becomes an integral part of the robot design; resources are spend on sensors and low-level control schemes. Since swarm robotics is heavily inspired by natural systems, and natural systems do not necessarily avoid collisions, we belief there is an increasingly growing narrative for research on collision-based swarm robotic behaviours. We will briefly cover the swarm robotic studies that focused on collisions. Kernbach et al. and Schmickl et al. worked on the re-embodiment of biological aggregation behaviours of honeybees. They show how to take advantage of collisions to develop scalable robot behaviours. In their work, swarm robots converge to light sources without requiring inter-robot communication. Concretely, they minimize sensing and computation by evaluating robot data only once per collision; more frequent collisions lead to more data evaluations [5][6]. Mayaa et al. harnessed collisions to help localise a robot within an arena. The arena was divided into differently sized segments, whereas each segment was inhabited by differently sized robot groups. Robots used

*This work was supported by the Centre of Doctoral Training in Intelligent Sensing and Measurement (EP/L016753/1) UK and Offshore Robotics for Certification of Assets (ORCA) (EP/R026173/1) UK).

[1]Markus P. Nemitz is a PhD candidate at the School of Engineering, Institute for Integrated Micro and Nano Systems, The University of Edinburgh, The King's Buildings, Edinburgh, EH9 3LJ, UK and Academic Affiliate in the Computer Science and Engineering Department, University of Michigan, Ann Arbor, Michigan, USA m.nemitz@ed.ac.uk nemitz@umich.edu

[2]Edwin Olson is Associate Professor in the Computer Science and Engineering Department, University of Michigan, Ann Arbor, Michigan, USA ebolson@umich.edu

[3]Adam A. Stokes is University Lecturer at the School of Engineering, Institute for Integrated Micro and Nano Systems, The University of Edinburgh, The King's Buildings, Edinburgh, EH9 3LJ, UK adam.stokes@ed.ac.uk

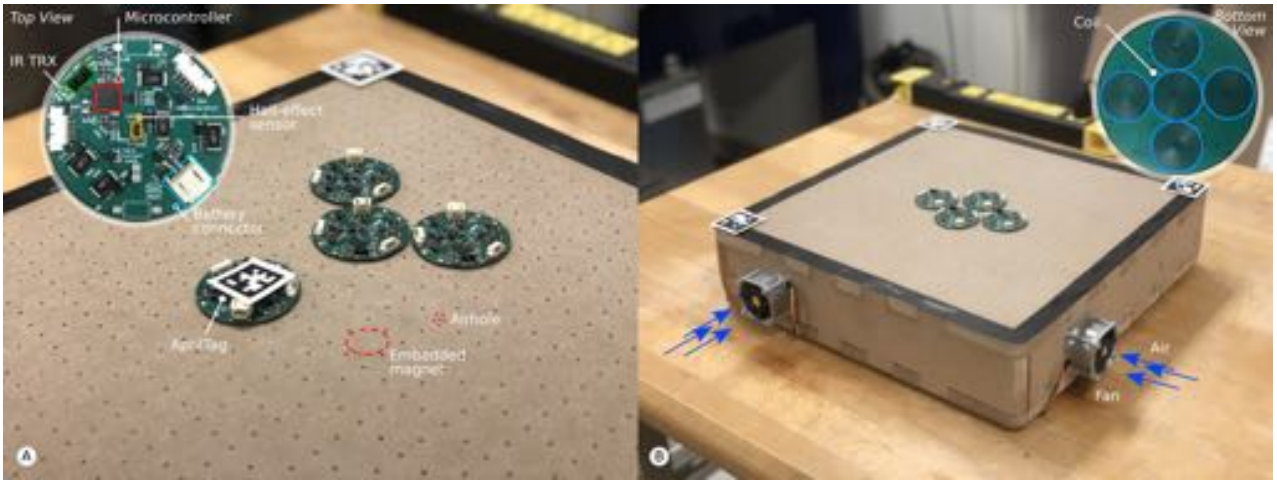


Fig. 2. The HoverBot system. A) The HoverBot is displayed in detail in the top left corner. It consists of a low-cost microcontroller, an infrared transceiver and a Hall-effect sensor. Permanent magnets are embedded into the platform and air holes are drilled through the surface as exemplary indicated through red circles. We placed AprilTags on a HoverBot as well as in three of the four corners of the magnet-levitation table. This setup allows us to keep track of HoverBot's position during experiments. B) The bottom side of the HoverBot is displayed in the top right corner. A HoverBot possesses five planar coils that it uses to manoeuvre two-dimensionally on the magnet-levitation table. We installed four fans, one on each side, to supply HoverBots with a constant airflow beneath their contact surface.

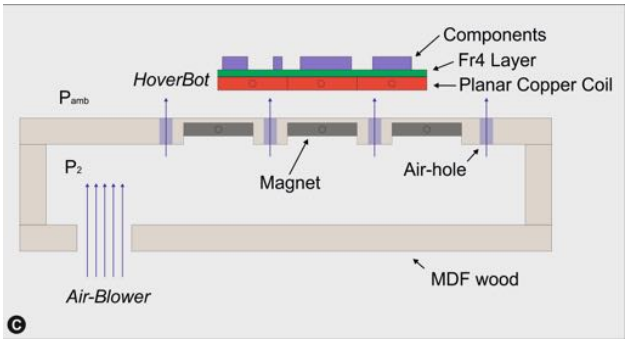


Fig. 3. Conceptual system overview. An air blower forces air into the magnet-levitation table creating a pressure differential between the inside and outside of the table. Air streams through the porous surface of the magnet-levitation table creating air-cushions beneath HoverBots which makes the robots levitate. HoverBots energise their planar coils and interact with the embedded magnets to move two-dimensionally.

collision detection as information source to determine their locations [7].

C. Approximating Collisions

In these studies, robot behaviours were collision dependent, however, physical collisions were still avoided. Robots detected close field objects with proximity sensors and accounted them for collisions. However, *true* collisions cause physical interactions amongst robots and their immediate environment; collision chains might even displace many robots at the time and possibly change the outcome of an experiment; approximating collisions neglects their physical impact on the real world.

D. Physical Collisions

The impact of a collision is dependent on the momentum of the robot $\vec{P} = m \times \vec{V}$, whereas fast velocities $\vec{V}$ or heavy masses m increase momentum. We consider scenarios in which robots move or rest. If two robots move and collide the total momentum is dependent on the velocity vectors of the robots. If one robot collides with a resting robot, the moving robot has to overcome the static friction of the resting robot to make it move.

Collisions are influenced by robot locomotion. In descending order starting with the most commonly used locomotion strategy, we look into the various strategies and discuss how they might influence collisions : i) wheeled locomotion ii) slip-stick locomotion iii) active low-friction locomotion [8]. Wheeled robots are faster and heavier than robots that use slip-stick or active low-friction locomotion, therefore their momentum is greater. However, wheeled robots are also more difficult to move due to their mass and corresponding static friction. Robots that use slip-stick locomotion are light and their velocities low causing small momentum which might be not sufficient to overcome the static friction of resting robots of their kind. Robots that use active low-friction locomotion are also light and their velocities (currently) low, however, they are visually frictionless. In this scenario, a collision between resting and moving robots results in movement as illustrated in Figure 1D. In the following section we review active low-friction locomotion and its first implementation, the HoverBot system.

II. The HoverBot System

A. Active Low-friction Locomotion

To move - on land, in water, or in the air - always requires an expenditure of energy. Reducing the resistance to motion, namely, friction, allows a greater range of travel for a given input of energy [9]. However, instead of enhancing locomotion, we enable locomotion by reducing friction.

HoverBot is a simple robot that is only capable of manoeuvring if it is supplied with a constant air flow beneath

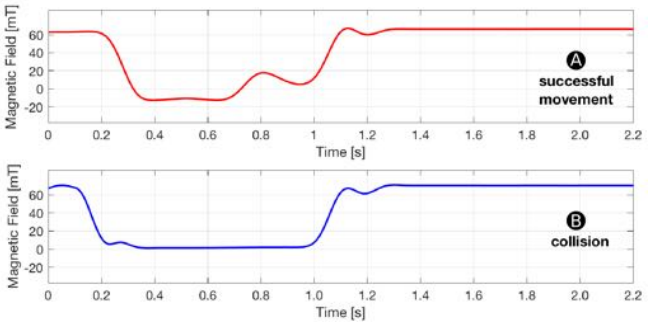


Fig. 4. Magnetic Field Profiles. These are examples of magnetic field measurements (signatures) measured by a HoverBot during movement and show A) successful movement and B) collision. The time series are distinct, they vary in time and magnitude.

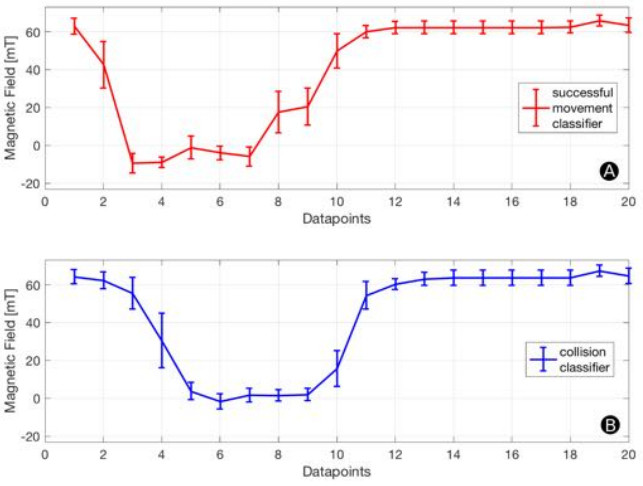


Fig. 5. Classifier Parameters. Each datapoint consists of a mean value and standard deviation. These values are stored in the microcontroller's memory and used for online classification.

its contact surface. HoverBot's working principle is shown in Figure 3. The air flow reduces the friction between robot and table allowing relatively weak forces to be used for locomotion. Specifically, we embedded permanent magnets into a levitation table. HoverBot possesses planar coils which interact with these permanent magnets, resulting in two-dimensional locomotion. Such forces would be insufficient if friction had not been reduced. This concept relaxes actuator boundaries allowing a significant simplification of the robot's actuation and control system. This locomotion strategy is called *active low-friction locomotion* and is further discussed in our publication [8].

B. The Magnet Levitation Table

The table supplies an air flow beneath HoverBot's contact surface creating an air cushion that reduces friction between robot and locomotion substrate. The differential pressure that is required to lift a HoverBot can be estimated by the following equation [10]:

$$\Delta P = (P_2 - P_{amb}) \geq \frac{M \times g}{\pi \times R^2}. \quad (1)$$

Equation 1 implies that an increase in robot weight M or a reduction of its surface area $\pi \times R^2$ can be encountered by an increase in differential pressure ΔP.

The permanent magnets that are embedded into the top surface serve a double purpose, they: (1) act as magnetic anchors that a HoverBot utilizes to maneuver and (2) give rise to a magnetic field with a discrete regular pattern of features which HoverBot is capable of sensing with its Hall-effect sensor. All magnets were assembled mono-directionally: north-pole facing up.

C. The HoverBot

HoverBot consists of a single four-layer Printed Circuit Board (PCB), shown in Figure 2, and a detachable 300 mAh lithium polymer battery. The bottom layer comprises five planar actuation coils. Each HoverBot has a diameter of 39 mm and weighs 19.4 g with, and 7.4 g without, a battery. HoverBot possesses a low-power microcontroller (Atmel's SAMD21E series), programming and debug ports, an infrared transceiver, a Hall-effect sensor, and a transistor circuit.

From the outset, the HoverBot system was *designed for manufacturability*: HoverBots only require electronics components that are surface mountable, only require connecting a battery to a robot as an assembly step, use low-cost actuators and associated circuitry, do not require actuator calibration and move precisely on a discrete grid. For more details, please refer to our publication [8].

III. Detecting Collisions with HoverBots

HoverBots mainly consist of glass-reinforced epoxy laminate (FR4) which makes them very robust and difficult to break. HoverBots effortlessly collide with objects or other robots. Sometimes a collision impacts the trajectory of a HoverBot. Figure 1 illustrates a series of demonstrations in which robots collided with one another. While HoverBots embrace collisions, they are also capable of detecting them. HoverBots possess a single Hall-effect sensor and they utilise the magnetic field readings that occur during their movement to detect collisions.

A. Event Dependent Magnetic Field Measurements

HoverBots hover on air cushions and pull themselves towards magnetic anchors that are embedded into the arena surface. When HoverBots move, they measure time-dependent magnetic fields. Amongst other, it is possible to associate successful movements and collisions with distinct magnetic field measurements (signatures). Figure 4 shows examples of collision and successful movement signatures; they differ both in time and magnitude.

B. Time Sequence Classification

We group signatures into classes (here: collisions and successful movements) and then use signal processing techniques to learn offline representations for each class. Offline representations are essentially averaged versions of

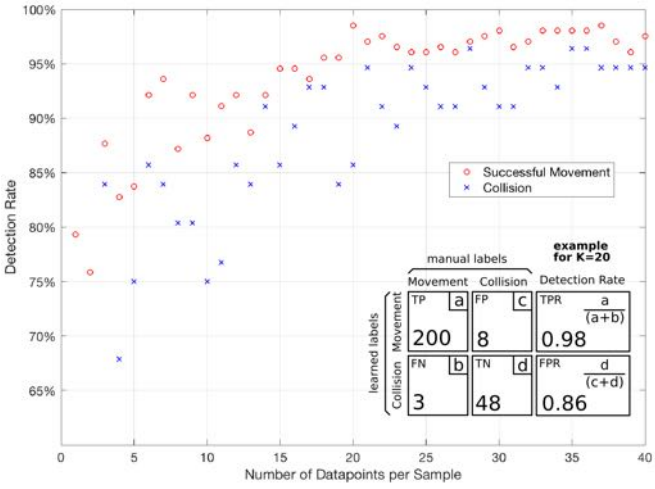


Fig. 6. The detection rate increases with the number of datapoints but starts stagnating once it exceeds 20. In the bottom right corner, we give an example of a confusion matrix for 20 datapoints. Legend: TP=True Positive, FN=False Negative, FP=False Positive, TN=True Negative, TPR=True Positive Ratio, FPR=False Positive Ratio.

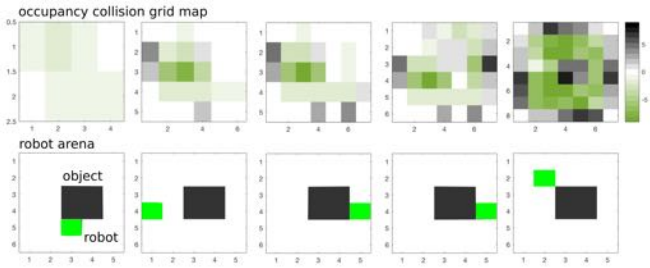


Fig. 7. A robot moves randomly in the environment; the bottom row shows its position, the top row its observations. A collision is accounted for by adding 1, a successful movement by subtracting 1 from its observation matrix. Positive accumulations are illustrated in gray-scale (collisions), negative accumulations in green-scale (successful movements). Over iterations, the arena object can be identified in the robot's observations. The robot uses a dynamically growing memory array to keep track of its observations; it is able to map environments without prior knowledge of their size.

signatures. While this averaging process is non-trivial when performed on variable-length signatures and might deserve an entire discussion by itself, this work presented here intends to give an overview of the HoverBot system, its capabilities and how it could serve the narrative of collision dependent robot behaviours. Therefore, we do not discuss the technicalities of the averaging process but refer to the key literature of our approach including our manuscript that is currently under review for publication [11][12][13].

Figure 5 shows the offline representations of the collision and successful movement classes having averaged a total of 259 signatures. The representations consist of a number of (mean μ - standard deviation σ) tuples, whereas the number of tuples is dependent on the number of datapoints per signature. HoverBot is capable of measuring dozens of magnetic field measurements per second, however, the magnetic field itself does not change that quickly. Once HoverBot measures a new magnetic field time series, it computes the Mahalanobis distances between the new measurements x_k and the representations μ_k σ_k for 1) collision and 2) successful movement, whereas fewer data-points $k \in K$ lead to less computation.

$$d(\mu_k, \sigma_k, x_k) = \sqrt{\sum_{k=1}^{K} \frac{(x_k - \mu_k)^2}{\sigma_k}} \qquad (2)$$

The Mahalanobis distance basically measures how many standard deviations σ is a point x away from the mean value μ [14]. We classify new measurements according to the minimum-distance class, which corresponds to the maximum-likelihood.

C. Detection Rate

The success rate of our classification is dependent on the number of data points per signature. While the detection rate increases with the number of datapoints, the detection rate starts stagnating once it exceeds 20 datapoints per signature. Figure 6 shows the detection rate as function of the number of datapoints per signature sample and gives a confusion matrix example for 20 datapoints. The successful movement detection rate is the true-positive and the collision detection rate the false-positive-rate of the confusion matrix. If signatures only contain a few datapoints, the corresponding class representations only contain a few datapoints too; the class representations lose their distinctiveness and the detection rate decreases.

HoverBots are capable of detecting physical collisions without requiring tactile sensors by analysing magnetic field measurements.

IV. Collision Mapping

While Kernbach and Schmickl et al's work is on collision-triggered *search* and Mayaa et al's work on collision-based *localisation*, we would like to hint briefly at the opportunity of using collisions for *mapping* environments.

A collision can indicate a dynamic (e.g. robot) or static (e.g. wall) obstacle. Robots can record collisions to build maps of their environment. The most trivial case might be a standard version of occupancy grid mapping in which robots know their pose, keep record of empty and occupied grid cells by detecting collisions, and store their observations in memory. For a better understanding, we performed a very basic simulation of a single robot that randomly collides with objects and builds a collision map which is illustrated in Figure 7. The simulated agent detects collisions with uncertainty; over time, the actual environment appears, the robot revisits cells and statistically detects more collisions correctly than wrong.

The combination of i) detecting collisions to infer information from the environment and ii) utilising the physical impact of collisions to push agents towards new solution spaces seems very useful for the development of new robot behaviours and the study of emergence. Active low-friction locomotion may play a unique role in collision research since

it facilitates a collision-friendly environment by eliminating frictional resistance.

REFERENCES

[1] W. Nachtigall, A. Wisser, *et al.*, *Bionics by examples*. Springer, 2014.

[2] W. Shao and D. Terzopoulos, "Autonomous pedestrians," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 19–28, ACM, 2005.

[3] J. Löber, F. Ziebert, and I. S. Aranson, "Collisions of deformable cells lead to collective migration," *Scientific reports*, vol. 5, p. 9172, 2015.

[4] D. Grossman, I. Aranson, and E. B. Jacob, "Emergence of agent swarm migration and vortex formation through inelastic collisions," *New Journal of Physics*, vol. 10, no. 2, p. 023036, 2008.

[5] S. Kernbach, R. Thenius, O. Kernbach, and T. Schmickl, "Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system," *Adaptive Behavior*, vol. 17, no. 3, pp. 237–259, 2009.

[6] T. Schmickl, R. Thenius, C. Moeslinger, G. Radspieler, S. Kernbach, M. Szymanski, and K. Crailsheim, "Get in touch: cooperative decision making based on robot-to-robot collisions," *Autonomous Agents and Multi-Agent Systems*, vol. 18, no. 1, pp. 133–155, 2009.

[7] S. Mayya, P. Pierpaoli, G. Nair, and M. Egerstedt, "Collisions as information sources in densely packed multi-robot systems under mean-field approximations," in *Proc. Robot., Sci. Syst. Conf.*, 2017.

[8] M. P. Nemitz, M. E. Sayed, J. Mamish, G. Ferrer, L. Teng, R. McKenzie, A. O. Hero, E. Olson, and A. A. Stokes, "Hoverbots: Precise locomotion using robots that are designed for manufacturability," *Frontiers in Robotics and AI*, vol. 4, p. 55, 2017.

[9] V. Radhakrishnan, "Locomotion: Dealing with friction," *Proceedings of the National Academy of Sciences*, vol. 95, no. 10, pp. 5448–5455, 1998.

[10] L. G. Leal, *Advanced transport phenomena: fluid mechanics and convective transport processes*, vol. 7. Cambridge University Press, 2007.

[11] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[12] M. Morel, C. Achard, R. Kulpa, and S. Dubuisson, "Time-series averaging using constrained dynamic time warping with tolerance," *Pattern Recognition*, vol. 74, pp. 77–89, 2018.

[13] M. P. Nemitz, R. Marcotte, M. E. Sayed, G. Ferrer, A. O. Hero, E. Olson, and A. A. Stokes, "Multi-functional sensing for swarm robots using time sequence classification: Hoverbot, an example," *Frontiers in Robotics and AI*, 2018, under review.

[14] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1–18, 2000.

Appendix H

Poster: Collective Perception in Multi Agent Systems

Conference: Centre for Doctoral Training in Intelligent Sensing and Measurement, Annual Conference 2015/16, United Kingdom

Author: Markus P. Nemitz and Adam A. Stokes

Online access to poster

Collective Perception in Multi Agent Systems

Markus Nemitz[1] and Dr. Adam A. Stokes[2]

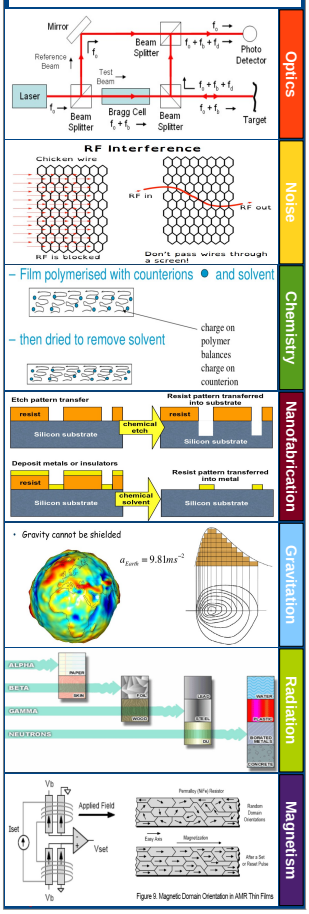
1- Ph.D. Candidate in Integrative Sensing and Measurement, University of Edinburgh, (m.nemitz@ed.ac.uk)
2- Principal Investigator, Stokes Research Group, University of Edinburgh, (adam.stokes@ed.ac.uk)

THE UNIVERSITY of EDINBURGH

2014 — Semester I — Semester II — Semester III — time

First year of integrated studies

Fundamentals



Entrepreneurship

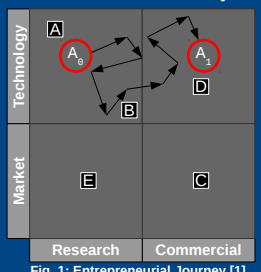


Fig. 1: Entrepreneurial Journey [1]

A Strong focus on long-term research on specific technology. The researcher probably focuses on fundamental scientific areas such as understanding the gene sequence of a specific strain of bacteria.

B Research focus on a specific technology but with some consideration of possible applications.

C Focus on commercial applications to meet specific market needs.

D Commercial role based on a specific technology embodiment.

E Research role designed to solve a customer problem without as much focus on the specific technology.

A_0 Initial condition: The journey encountered numerous reversals and directional changes. The initial direction is often the innovator's technology affinity.

A_1 Final outcome: Position on the journey map showing the inventor's new role identity.

Modular Robotics for Sensing Applications

Modular robots differ from traditional robots in that their entire robot body is made from a collective of individual sub-modules.

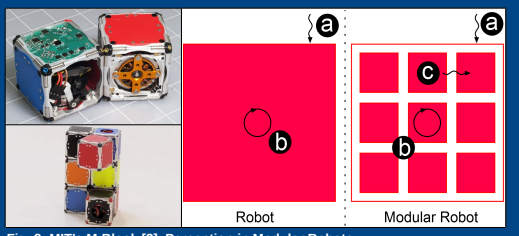


Fig. 2: MIT's M-Block [2], Perception in Modular Robots

a Exteroception
b Proprioception

a Exteroception
b Proprioception intra-modular
c Proprioception inter-modular

We subdivide perception into inter-, intra-proprioception, exteroception and accordingly **classify** existing modular robotic systems.

Operations	Requires Sensors
Self-reconfiguration	No
Self-assembly	Yes
Self-disassembly	No
Self-repair	Yes
Flow	No
Gait	No
Self-adaption	Yes
Grasping	Yes
Enveloping	Yes
Collective-actuation	No

Tab. 1: Possible Operations of a Modular Robot

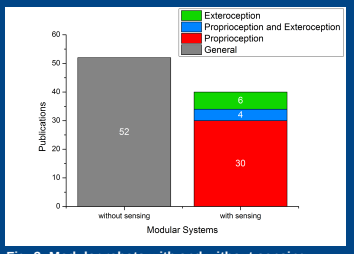


Fig. 3: Modular robots with and without sensing

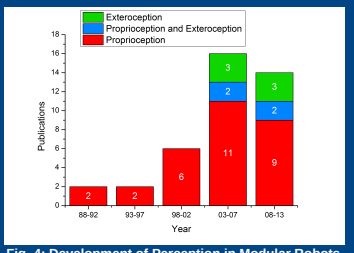


Fig. 4: Development of Perception in Modular Robots

2015

Three years PhD

Collective Perception in Multi Agent Systems

Fig. 5: Swarm Examples [3]

Characteristics of Swarms
- Local Sensing
- Local Communication
- Autonomy

Research Aim
- Heterogeneous Swarms
- Variety of Local Sensing Modalities
- Decision Making
- Local and Global Data
- Collective Perception

Applications
- Cooperative Environment Monitoring
- Moving Target Localization and Tracking,
- Convoy Protection, Search and Rescue
- Exploration, Health Systems ...

Biology — Computer Science — Mechanical and Electronics Engineering — Data Science

Experimental Platform

Fig. 6: Kilobot [4]

Simulations

Fig. 7: Local / Global Behaviour [5]

CDT-ISM cohort

CDT Colleagues/friends in
- Chemistry
- Physics
- Biology
- Engineering

An entire cohort that simultaneously tackles similar Ph.D. challenges

Access to the University of Edinburgh and the University of Glasgow

Internal CDT organisations such as the CDT-ISM Entrepreneurship Journal Club

Continuous progress meetings with Glasgow and Edinburgh academics.

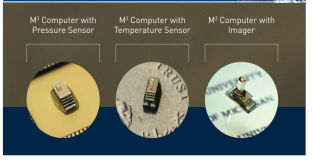
Michigan Collaboration

Prof. Alfred Hero
Data Science

Prof. Edwin Olson
Multi-autonomous Systems

Prof. Mingyan Liu
Sensor Networks

Prof. David Blaauw
Integrated Circuits and VLSI Design



2018

[1] Inventing Entrepreneur, Adam J. Bock
[2] M-Block, MIT, Daniela Rus, 2013
[3] Science of Swarms, wired.com, 2013
[4] Kilobot, Harvard University, Radhika Nagpal, 2014
[5] glswarm - Swarm Simulation and Visualization

Acknowledgments to the EPSRC and the Centre for Doctoral Training in Integrative Sensing and Measurement.

EPSRC Engineering and Physical Sciences Research Council — ISM Integrative Sensing and Measurement — University of Glasgow — THE UNIVERSITY of EDINBURGH

Appendix I

Poster: Distributed Sensing with Swarm Robots

Conference: Engineering Graduate Conference at the University of Edinburgh 2017

Authors: Markus P. Nemitz, Edwin Olson, Alfred O. Hero, and Adam A. Stokes

Online access to poster

Distributed Sensing with Swarm Robots

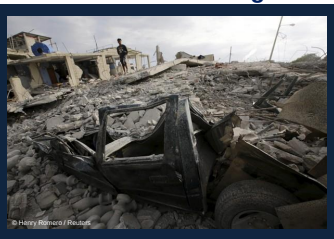
Markus P. Nemitz[1], Prof. Alfred O. Hero[2], Prof. Edwin Olson[3] and Dr. Adam A. Stokes[4]

1- Ph.D. Candidate, CDT in Intelligent Sensing and Measurement, The University of Edinburgh, (m.nemitz@ed.ac.uk)
2- Principal Investigator, Stokes Research Group, The University of Edinburgh, (adam.stokes@ed.ac.uk)
3- Research Collaborator, Hero Group, University of Michigan, (hero@eecs.umich.edu)
4- Research Collaborator, APRIL Laboratory, University of Michigan, (ebolson@umich.edu)

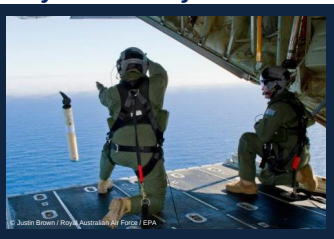
THE UNIVERSITY of EDINBURGH

COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY of MICHIGAN

Motivation: Performing search quickly and reliably ①

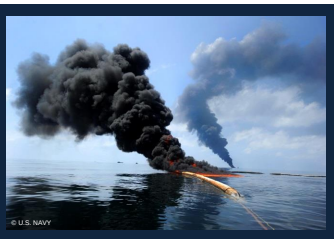


Finding survivors after a natural disaster (e.g. Nepal in 2015).



Finding a wreck after a plane crash (e.g. MH370 in 2014).

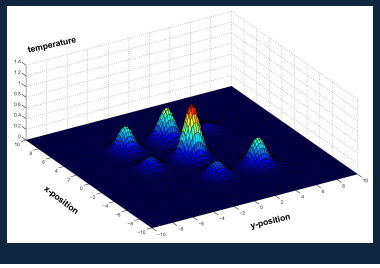
Finding a route through contaminated plants (e.g. Fukushima in 2011).

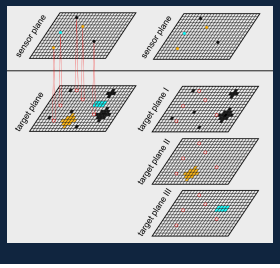


Finding oil spills in the ocean (e.g. Deep Water Horizon in 2010).

Model Description ②

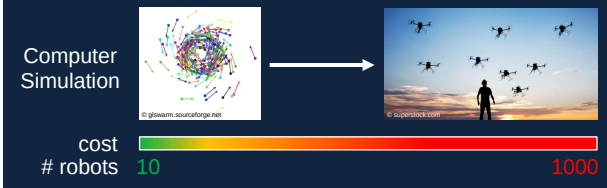
Each target can be described through a set of sensing modalities. For example, a survivor of a natural disaster could be detected by utilising temperature, sound and visual sensors.





A Step Towards Real World Applications ③

Developing swarm behaviours in-silico and then programming the final swarm (>1000) remains an impractical approach. The cost and effort to conduct such experiments is currently exorbitant.



However, computer simulations must be validated since their fidelity to the real world is difficult to verify or improve without feedback from physical experiments. An embodied simulation is a low-cost intermediate step between computer simulation and real world application.

Embodied Simulations ④

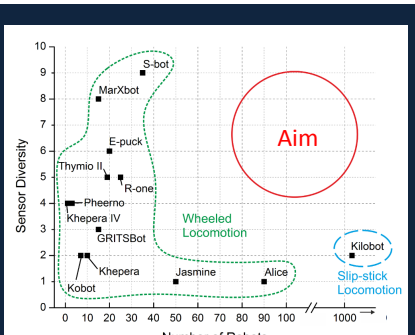
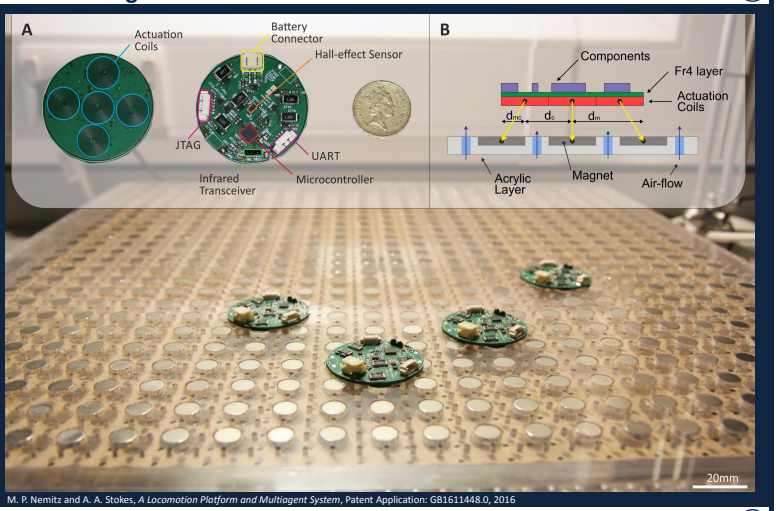


TABLE II
COMPARISON OF WHEELED, SLIP-STICK AND LOW-FRICTION LOCOMOTION

Feature	Wheeled [18]	Slip-stick [12]	Low-friction
Robot Velocity	$25\frac{cm}{s}$	$1\frac{cm}{s}$	$1.2\frac{cm}{s}$
Rotational Velocity	$820\frac{deg.}{s}$	$45\frac{deg.}{s}$	No Rotation
Motion	Continuous	Continous	Discrete ($5mm$)
Battery Life	$30min$-$5h$ @150mAh	$3h$-$24h$ @160mAh	$40min$-$600h$ @300mAh
Dependencies	No	Shiny Surface	Levitation Table
Hardware Odometry	Stepper Motors	No	Hall-effect Sensor
Actuator Calibration	Not Required	Required	Not Required
Non-SMD Components	≥ 4	≥ 5	0
Size	$30mm$	$33mm$	$39mm$
Cost[1] @ 1000 Units	$12.34	$3.12	$1.96

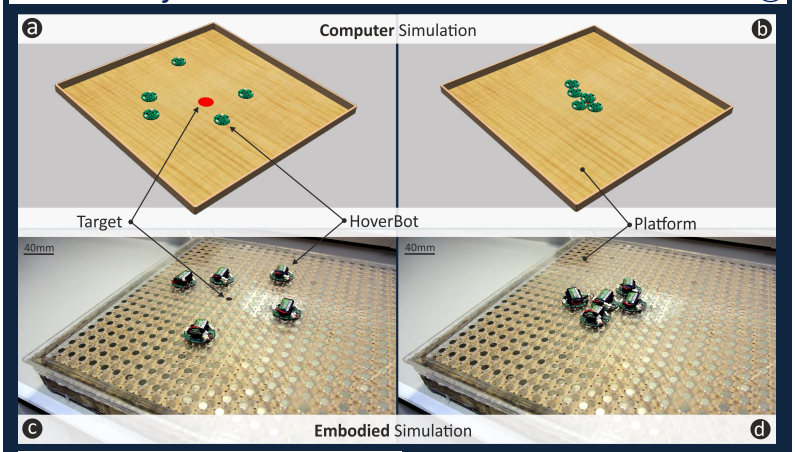
Nemitz et al. *Low Friction Locomotion: A Functional Locomotion Strategy for Swarm Robots at Low-cost*, IROS 2017 (submitted)

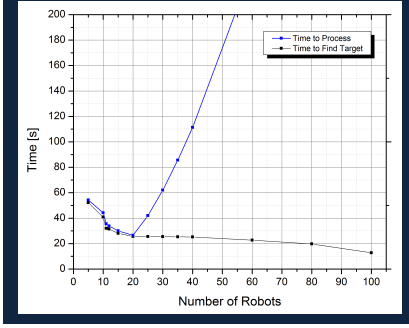
Levitating Circuit Boards ⑤



M. P. Nemitz and A. A. Stokes, *A Locomotion Platform and Multiagent System*, Patent Application: GB1611448.0, 2016

Preliminary Results ⑥





Target Modality: Magnetic field
Communication: Infrared light
Algorithm: Random search
Number of robots: 5
Table size: 500 x 500 mm

Results: The computer's time-to-process the simulation increased quadratically with an increase of robots. An increase from 20 to 60 robots did not considerably expedite the time-to-find-target.

Acknowledgements ⑦

I would like to thank Sayed Mohammed (PhD student, Stokes Research Group, The University of Edinburgh) for his helpful support on conducting the computer and embodied simulation experiments.

EPSRC Engineering and Physical Sciences Research Council
Scottish Funding Council
University of Glasgow
ISM Integrative Sensing and Measurement
april robotics laboratory

Appendix J

Poster: HoverBots: Embracing and Detecting Collisions Using Robots Designed for Manufacturability

Conference: International Conference on Robotics and Automation 2018

Workshop: Swarms: From Biology to Robotics and Back

Authors: Markus P. Nemitz, Edwin Olson, and Adam A. Stokes

Online access to poster

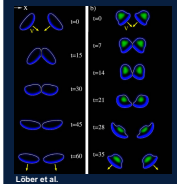
HoverBots: Embracing and Detecting Collisions Using Robots Designed for Manufacturability

Markus P. Nemitz[1,2] Edwin B. Olson[2] Adam A. Stokes[1]

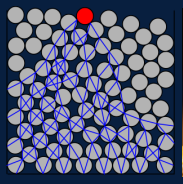
[2]School of Engineering, The University of Edinburgh

[1]Computer Science and Engineering, University of Michigan

Collisions are natural, robots tend to avoid them



Cells



Granular media

Bees

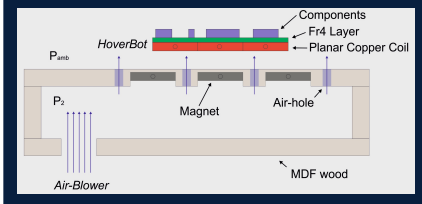
Fish

Human crowds

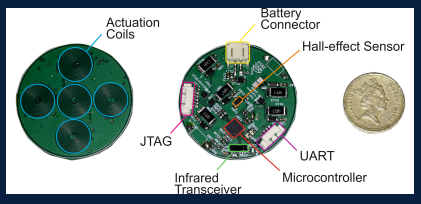
Embracing collisions is natural: the vast majority of physical, biological, and chemical agents collide. Collisions happen frequently in nature, at different places, and under various circumstances.

Robots avoid collisions: on the contrary, the vast majority of (swarm) robotic systems avoid collisions. Roboticists spend significant resources (sensors, max. priority processing, etc.) on control schemes to prevent collisions.

Swarm robots that embrace collisions: HoverBot, an Example

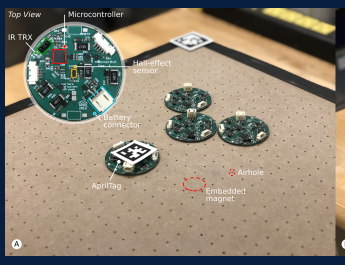


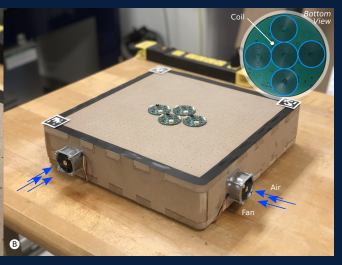
Active low-friction locomotion is a new locomotion strategy for swarm robots. We developed active low-friction locomotion to: i) enable robots to embrace collisions; and ii) to eliminate robot manufacture. Robots that use active low-friction locomotion require electro-magnetic actuators and a specialised arena.



HoverBot is the first implementation that uses active low-friction locomotion. HoverBots possess a Hall-effect sensor, an infrared transceiver, and a low-cost microcontroller. HoverBot's actuation system consists of an array of planar coils and a transistor circuit. HoverBots consist of a single PCB and a battery. They can be ordered as populated PCBs – no manual manufacture required.

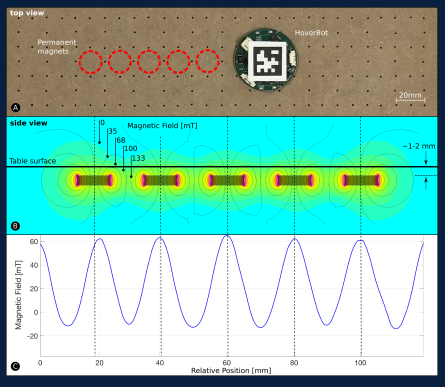
HoverBots operate on a **magnet-levitation table.** The magnet-levitation table supplies HoverBots with: i) a constant air-flow beneath their surfaces; and ii) permanent magnets which serve as magnetic anchors. HoverBots move by taking advantage of their low-friction environment and by actuating their planar coils to interact with the magnetic anchors.





@ Nemitz et al. "HoverBots: Precise Locomotion Using Robots That Are Designed for Manufacturability." *Frontiers in Robotics and AI* 4 (2017)

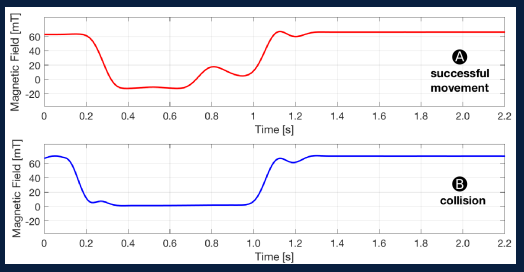
Detecting collisions with a single Hall-effect sensor



Permanent magnets are embedded into the magnet-levitation table. They evoke a position dependent magnetic field.

We discovered that HoverBot's magnetic field measurements can be associated to events. For example, the magnetic field reading of a successful movement looks distinctly different than the magnetic field reading of a collision.

We can use a time series classifier to discriminate between the various magnetic field curves (signatures) i.e. events.



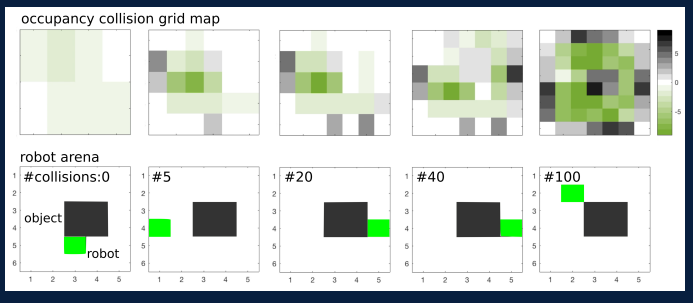
While this method has only been applied to HoverBot's magnetic field measurements, any signature (sensor modality independent) can be detected through a time series classifier if the signature fulfills the classification criteria. Classification criteria: signatures must be: time variant; ii) systematically reoccurring; and iii) distinct to other signatures of the classifier.

@ Nemitz et al. "Multi-Functional Sensing for Swarm Robots Using Time Sequence Classification: HoverBot, an Example." *Frontiers in Robotics and AI* 5 (2018)

Utilising robot collisions to map environments

Collisions can be utilised as information sources. We trained our classifier to detect static objects; inter-robot collisions were explicitly excluded from the training set. However, the classifier can be trained with any signatures as long as the signature fulfills the classification criteria.

Our current research focuses on occupancy collision grid mapping. Occupancy grid mapping is a very basic mapping technique. Robots create maps of their environment from sensor measurements (here: collisions). Our collision detection is probabilistic; the more often robots collide with static obstacles, the more we learn about the environment; the number-of-collisions-workload can be shared amongst robots to fasten the task.



EPSRC Engineering and Physical Sciences Research Council

Scottish Funding Council

University of Glasgow

ISM Integrative Sensing and Measurement

april robotics laboratory