Aalborg University
Department of Electronic Systems
Fredrik Bajers Vej 7
9220 Aalborg Ø
Telephone +45 99 40 86 00
http://es.aau.dk/

Title:
Robotic Swarm - Development and Formation

Theme:
Modeling and Control

Specialization:
Intelligent Autonomous Systems

Project period:
9th and 10th Semester
Spring 2009 to Fall 2010

Project group:
09gr937@es.aau.dk

Group members:
Simon Bjerg Mikkelsen
René Jespersen

Supervisors:
Trung Dung Ngo

Copies: 4

Pages: 63

Attachments: 1 CD

Finished 03/06/2010

Synopsis:

A large pensum of related papers was studied by a reading group to gain knownledge in the field of Swarm Robotics. The aim this study was to increase the focus of the project to formation of swarm robots which is expressed in the development of a formation algorithm.
Methodology presented A unified methodology for developing swarm robots was presented, solving the practical problems scientists faces when working with real world robots. The paper was submitted and accepted on the RISE10 conference in Sheffield, England, where both group members presented their work.
A large Simulink model was developed able to simulate the behaviour of 10 robots in a swarm. An formation algorithm using potential field forces was implemented able to create an attractive force from communication and a repulsive force from the measuring of light intensity. From using both communication and light intensity it was possible to control both the distance between the robots and a common phase alignment.
The development of 10 lightweight swarm robots was represented. Based on an old version using infrared sensors to navigate and communicate, a new mechanical chassis was created and an additional sensorboard able to broadcast was attached. Furthermore was the motor controller-board replaced with a new designed board providing both a radio and a bluetooth module. An simple implementation of the algorithm from simulation, proved the concept by controling the robots to the expectet behaviour.

Preface

This report documents the project, conducted at Aalborg University in the period from September 2009 to June 2010 with the titel: Robotic Swarm - Development and Formation. 10 swarm robots was provided from previous projects but after an analysis they where found insufficient for the scope of this project. For that reason it was decided to develop 10 new robots usable as a testplatform within the scope of this project aswell as future research. The robots was decided to be produced as proffessional as possible. The chassis was produced at the metal workshop by the university and to produce the boards a company in China was used.

The report is split into three main parts: Analysis, Development, Test and Conclusion. Furthermore is an Appendix presented illustrating and discribing additional supportive content. The Analysis part presents the outcome from papers presented and discussed at a readinggroup consisting of the two groupmembers and the superviser. A description of the system available in form of an arena and provided robots is presented. Through the description it is analysed to point out usable functionalities and the problems of the system. Furthermore it introduces the goal of the project which is inherited to requirements illustrated in a Test Specification.

The Development part written untraditionally since it is based on three individual papers, each discribing a part of the project. Because they are individual there are content that are repeated multiple times. The first paper in Chapter 4 is presenting a unified methodology to overcome the practical challenges in the evolution of developing swarm algorithms. This paper was accepted at the Risky Intervention and Surveillance conference held at the university of Sheffield in Feburary 2010. Both groupmembers participated in the conference as the only students to present the paper. The next paper in Chapter 5 presents a method to create formation using attractive and repulsive forces created from both light intensity and communication. The method is supported by results from simulations. This paper is ready to be submittet to a suitable conference in the nearest future. The third paper in Chapter 6 is a technical written paper and are presenting the development of a set of new swarm robots. Furtermore is an implementation of the method from the simulation on real world robots presented. The implementation is supported with results from simple proof of conpept tests of the real world robots. For the paper to be ready for submission the implementation of the method, needs to be updated to get good results as in the simulation paper.

A CD is appended with the report in PDF format, videoclips of the robots running, software for the three boards as it is implemented at the time of writing, papers discussed in the readinggroup, the simulink model of the simulation and the Orcad drawings of the designed boads.

Sources are indicated with the first three letters of the author's last name and year followed by page numbers, if referring to a book e.g. [Law97, p.38]. Internet sites are not marked with page reference. Figures, formulas and tables are numbered by chapter and location, e.g. the second figure in chapter three is numbered 3.2. Functions are written as `main()` and the same font is used for file names and variables. Abbreviations and definitions used in the project are presented after this preface.

The authors would like to thank Professor Trung Dung Ngo for supervising.

Aalborg University, June 3rd, 2010

Simon Bjerg Mikkelsen

René Jespersen

Contents

Part I
Analysis

Chapter 1

Introduction

Ever since the beginning of life, animals and creatures have been developing their behavior, appearance and capabilities by adapting to the environment they live in. The knowledge from that development through million of years, is very interesting from the perspective of an engineer. It has been a great inspiration for humans for thousands of years. The Wright brothers were the first humans to fly successfully, helped by a plane constructed from the inspiration of birds. Birds have also inspired to aero design and the swarm behavior of a flock of birds forming a V-shape to reduce the air resistance to gain a lift. Hence the individual bird is using less energy to travel a longer distance than it would do on its own. Another example is ants standing on top of each other to create a bridge for the colony members to cross over gabs much longer than their individual body length. Swarm behavior is interesting for humans for multiple reasons. The main reason is that in a swarm, many simple homogeneous individuals can meet greater capability when working together, than the individuals themselves are able to achieve. Furthermore a swarm is both flexible and robust in the sense that it is possible to add or remove a member of the swarm, without any major faults such as deadlocks or lack of capabilities. Hence there is no leader or specialists in the swarm, and thereby no single point of failure.

Parallels can be drawn from the biology behavior to robots. Big complex industrial robots can be very expensive, it is very interesting to see whether the swarm collaborative methods can be applied to applications as an alternative to these. The advantages are the same as from the biology, and smaller simple robots are cheaper to produce and thereby easier and cheaper to scale the production up and down, as the task demands it. Furthermore the swarm is able to replace a broken member by another automaticly, since the robots are homogeneous. Hence it has no single point of failure, resulting in savings in a breakdown.

An example of an application for the swarm method could be in hazardous environments like a rescue recovery or in warzones etc. When a building has crashed or is in any way too dangerous to enter for humans, a set of swarm robots could search for surviving people using formation. By use of small simple robots, the loss of a faulty member would not be cost full and since there is no leader it will not affect the overall mission, because all other robots can continue.

The use of swarm collaborative methods is a fairly new subject in robotics and has not really met the market yet. The testing of these methods are mainly done in simulations, since testing on real world robots is very time consuming and costly. We will start our development by looking into which project is already presented to learn what have been done. This will help us to solve problems and improve the solution. The projects we found interesting is described in the following section.

1.1 Related Work

To analyse previous work that are related to the topic of formation using robotic swarms a reading group was created between the two groupmembers and the superviser. A list of the papers found relevant was divided into three different topics; Multi-robot Systems, Swarm Algorithms and Formation Control. The two groupmembers read one paper each to present for the readinggroup. Followed by a discussion of the relevance and usabillity of the methods presented in each paper. The essential parts of each paper are now discribed.

1.1.1 Multi-robot Systems

Editorial: Advanced in Multi-robot Systems [APP02]

This paper discribes the field of distributed robotics. They devide their research in seven different topics interesting within multi-robot systems.

- Biological inspirations
- Communication
- Architectures
- Localization, mapping and exploration
- Object transport and manipulation
- Motion coordination
- Reconfigurable robots

This paper provides an overview of the study of multi-robot systems and is interesting when first started to make reseach in this area to gain knowledge on the common topics.

Cooperative Mobile Robotics: Antecedents and Direction [CFK97]

This paper is a survey over the trends for research in the field of cooperating mobile robots. Since it is from 1997 a lot has happened since. It does however give a good basic understanding of the field of cooporaion for robots. The areas of interest are:

- Group Architecture
- Resource Conflict
- Origin of Cooperation
- Learning
- Geometric Problems

Distributed Intelligence: Overview of the Field and its Applications in Multi-robot Systems [Par08]

The goal for this paper is to create overview of the concept behind distributed intelligence. To classify the different type of distributed intelligence the types of interactions that can take place between the robots in the system was defined through a set of parameters.

- Type of goals: Can either be an individual goal or shared for the whole system.
- Awareness of others: Either the robot is aware of others or not aware.
- Actions advance goal for others: The action of the individual robot helps the others to advance towards their goal.

From these parameters four different types of interactions was defined illustrated in Figure 1.1.

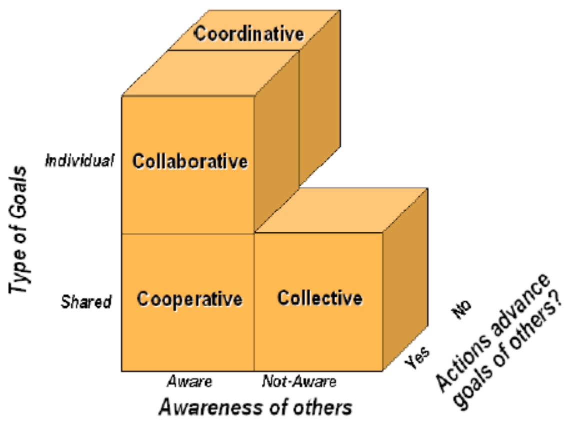


Figure 1.1: Figure from the paper illustrating the coherence between the parameters and the interactions

- Collective: These robots are not aware of other robots but share a common goal and its actions help that common goal.
- Cooperative: These robots are aware of other robots and share their common goal and its actions help that common goal.
- Collaborative: These robots are aware of other robots but have individual goals but its action still helps the common goal.
- Coordinative: These robots are aware of other robots but have individual goals and its action does not help the common goal.

By understanding and defining the types of interactions desired in a distributed intelligence system, it provides insight to achieve an appropriate solution strategy. A set of three paradigms where presented as solutions to achieve these interactions.

- The bioinspired paradigm: Based on the observations of animal behaviour, parrallels to multi-robot systems can be drawn. Many animals appear to be hardwired to specific operations producing very stereotypical reactions to particular stimuli, just as the input/output relation on robots. When a flock of birds are flying they are forming a v-shape to reduce the airresistance. Parallels can be drawn to the use of large numbers of identical robots that individually have very little capability, but when combined with others can generate seemingly intelligent group behavior.

- The organizational and social paradigms: Based on organizational theory derived from human systems. Knowledge from the fields of sociology, economics, and psychology, and related areas, have proven valuable for understanding how to create systems of intelligent members that can work together to solve complex problems. In these approaches, robot interactions are designed by modeling individual and group dynamics as part of an organization. These approaches reduce the communications requirements among robots by making use of models drawn from these fields.

- The knowledge-based, ontological and semantic paradigms: The focus in these approaches is on knowledge sharing between heterogeneous robots, with the objective of easily allowing these entities to share and understand knowledge from different sources. Such approaches require a language for representing knowledge and is not often used in multi-robot systems. This is because physical robot systems are more challenged by noise and uncertainty in sensing and actuation, as well as low-bandwidth communications, limited power, and limited computation.

They examine how each of the paradigms discussed, would handle the multi-robot task allocation problem.

- The bioinspired paradigm: This paradigm usually assumes a large number of homogeneous robots that are all interchangeable. Any available robot that senses the need for a task to be done can select to perform that task, hence the task is allocated to that specifik robot. Stigmergy is a mechanism of indirect coordination between robots or actions. The principle is that the trace left in the environment by an action stimulates the performance of a next action, by the same or a different robot. Because of that robots do not have to explicitly communicate to decide which task to select.

- The organizational and social paradigms: This paradigm take use of different roles defining several specific tasks. The robots each select a role that are best suited for their capabilities, meaning that there are no need for homogeneous robots, but different functionalities can be present.

- The knowledge-based, ontological and semantic paradigms: This paradigm uses a modeling of the capabilities of the other members in the system. The model is created by collecting relevant task statistics by observing the quality of other members performance while solving tasks. Robots then use these models to select tasks to perform, which benefits the group as a whole.

The conclusion on the paper was that they where able to outline aspects of the field of distributed intelligence with the focus on the types of interactions that can occur in such systems. Some paradigms used to achieve distributed intelligence was presented and the choice of paradigm is not always obvious, because it depends upon the requirements of the application to be addressed.

Reactivity and Deliberation: A Survey on Multi-robot Systems [INS01]

Survey paper. It proposes a taxonomy on Multi Robot Systems as shown in figure 1.2. This taxonomy describes different levels of cooperation and also distinguishes between reactive and social deliberative behaviors. Reactive behavior is compared to instinctive reactions whereas in the social deliberate behavior the group of robots adapt their group behavior to an environmental change.

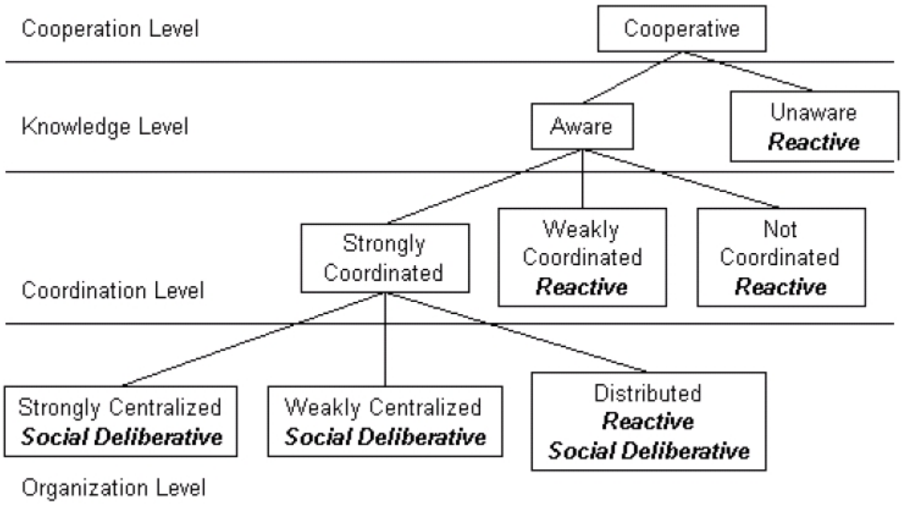


Figure 1.2: *Proposed taxonomy of Multi Robot System*

A Distributed Boundary Detection Algorithms for Multi-robot Systems [MD]

The main goal of this paper is to detect the boundary of a swarm based on a two dimensional configuration. It is desired to detect the boundary of the swarm as it is presented, eventhough some robots have a concave angle to the neighbours and other robots have convexity angle to the neighbours. Furthermore it is desired to capture the interior voids which can occur. A set of requirements was presented:

- The boundary form a connected subgraph of the robot network, so that messages can be routed around it and its properties can be estimated.

- The algorithm be fully distributed and require only local network geometry.

The local network geometry sensing is a model which combine the network connectivity of the robot and the positions of its neighbors measured relative to itself. The model is a practical compromise between a each robot knows only distances or connectivity to nearby robots, and a global coordinate system. Hence they assumpt that knowledge of local network geometry exist.

By using these informations, their algorithm classifies all the robots as either boundary or interior within one round of communication. For a rapidly changing network as most groups of robots creating a formation are, this is fast enough to react on changes. By using the local boundary classification they create a subgraph representing either an interior void or exterior boundary.

Their study have 112 small swarm robots available but a typical experiment uses 30-50 robots. Each of their swarm robots has a localization system providing the robots position and by using infrared communication able to determine their relative position to each other. On the top are three lights able to indicate wheter the robot is a part of either the interior, exterior boundary, interior boundary or a local articular point. Furthermore have each robot a radio used for programming the robots with software and for datacollection. They present great results in both simulation and real world examples.

1.1.2 Swarm Algorithms

Swarm Robotics and Minimalism [Sha]

This is a survey style paper. It proposes to make subclasses of robotic swarms to ease the distinction between different trends in research. Figure 1.3 shows the subclasses of robotic swarms.

- Scalable Swarm Robotics. The emphasis is on decentralisation, local control and communication, such that the approach is scalable.
- Minimalist Swarm Robotics although scalable due to its reliance on local communication and control also involves a commitment to the use of simple robots.
- Nature-inspired minimalist swarm robotics Constraining individual robots in a swarm robotic system to the abilities of social insects, provides an upper limit on their abilities, and also means that the approach remains scalable.

Sub area	Scalable	Minimalist	Nature-inspired constraints
Scalable SR	✓		
Practical Minimalist SR	✓	✓	
Nature-inspired minimalist SR	✓	✓	✓

Figure 1.3: *Subclasses of Robot Swarms (RS)*

Swarm Robotics: From Sources of Inspiration to Domain of Applications [Sah05]

The main topic of this paper is the coordination of a large number of simple robots and provide a definition of the swarming approach. This is done by:

- Describing the desirable properties of swarm robotic systems, as observed in the system-level functioning of social insects.
- Proposing a definition for the term swarm robotics, and putting forward a set of criteria that can be used to distinguish swarm robotics research from other multi-robot studies.
- Providing a review of some studies which can act as sources of inspiration, and a list of promising domains for the utilization of swarm robotic systems.

Based on the studies of social insects it is clamed that syncronized behaviour of the insects have no centralized coordination. Even though that is true, the swarm is both robust, flexible and scalable. These three characteristics are all disired in a multi-robot system. They discribe the characteristics as:

- Robustness is the ability to continue the wanted task eventhough one or more robots of the swarm fails while trying. To achieve robustness four things are important. Redundancy in the system using multiple robots capable of take over from a broken robot. Furthermore is a decentralized coordination desired since it provides no single point of failure of the system. Simplicity of the robots, gives less risk of failure compared to a single complex robot. Multiplicity of sensing increases the chance of correct sensing in the system.

- Flexibility is achieved when the system is able to generate modularized solutions to different tasks. Swarm robotic systems should also have the flexibility to offer new solutions of the tasks by applying different coordination strategies in response to the changes in the environment.
- Scalability requires that the goal of the tasks are achievable with different sizes of the swarm.

From these characteristics a definition for the term: *swarm robotics* is proposed:

Definition 1. Swarm robotics is the study of how large number of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among agents and between the agents and the environment.

For the definition to hold, a set of criterias to support the definition are presented:

- Autonomous Robots
- Large Number of Robots
- Few Homogenous Groups of Robots
- Relatively Incapable or Inefficient Robots
- Robots with Local Sensing and Communication Capabilities

The paper concludes that they have defined the swarm robotics in a new aproach to the control and coordination of multi-robot systems. The definition gives a good overview of the large field of swarm robotics.

Social Potential Fields: A Distributed Behavioural Control for Autonomous Robots [BC]

The focus of this paper is on very large robotic swarms and the term Very Large Scale Robotic(VLSR) is suggested to describe systems with hundreds to several thousand robots. The paper proposes using "Social" grouping and distributed control where the social groups have different behavours such as guarding and clustering. Each robots control is determined by the social group or groups it is a part of and the environment it can sense. The control is calculated locally where just the social relations are determined in a central controller. All control is implemented as dynamic virtual potential fields using the inverse power law or spring laws depending on which behavior is wanted. All data is from simulation. The paper is from 1999 so a lot has happened since it was written.

Algorithms for the Analysis and Synthesis of a Bio-Inspired Swarm Robotic System [BHKP]

The inverse problem is the focus of this paper. That is the problem of the design of individual behaviors to achieve a desired macroscopic behavior for the group. They propose a methodology for the design of individual behaviors to achieve a desired macroscopic behavior for a group. They use an example of an emigrating ant colony to decribe different abstraction levels of a swarm. This is shown in figure **??**

Distributed Sensing and Data Collection via Broken Ad hoc Wireless Connected Networks of Mobile Robots [Win]

This paper presents the development of an ad hoc wireless network useable for applications in distributed mobile robotics such as swarm robots. In a physically bounded region, a set of autonomous robots are

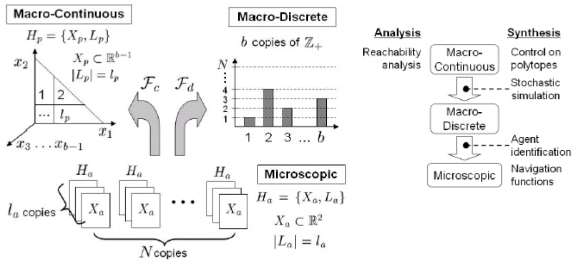


Figure 1.4: *(a)Levels of abstraction of a swarm; (b) Analysis and synthesis methodologies.*

deployed while reading sensory data from the surroundings and sending it back to a single collection point. The swarm of robots are decentralised, meaning there are no overall command or control structure. Each robot is equipped with a low range wireless network interface allowing the robot to communicate with its nearest neighbours. An ad hoc scheme of wireless networking is employed on the system. The method is able to collect data from the whole region eventhough not all robots are able to communicate with the collection point directly. By utilising the mobility of the robots to overcome the lack of global wireless connectivity it creates an ad hoc network usable to send collected data from all the robots through the network to the collection point.

The paper presents simulation data illustrating that the delay of collecting data is dependen on the numbers of robots deployed. Hence the conclusion of the paper is that the method is suitable for systems where realtime data is not required and some loss of data is acceptable.

Self-organizing Flocking in Mobile Robot Swarms [TCGS08]

This paper investigates how adding an alignment controller to a virtual potential field controller can make formations more compact. The aligment controller turns the robot to the average direction of its neighbours. It is based on real world experiments on 7 robots. The alignment is measured using a compass module on each robot and distances for the virtual potential field are measured using 8 IR sensors spaced 45°around the robot. They compare the entropy of formations from 4 different controllers given the same initial setup. The 4 different controllers are:

1. Virtual potential field with constant speed
2. Virtual potential field with modulated speed
3. Virtual potential field and alignment controller with constant speed
4. Virtual potential field and alignment controller with modulated speed.

They conclude that number 4 gives the lowest entropy i.e the most tight formation and the lowest spread in inter robot distances.

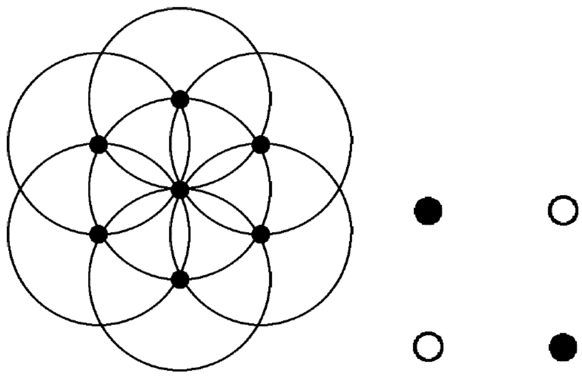
1.1.3 Formation Control

Distributed, Physical-based Control of Swarms of Vehicles [SSHH02]

This paper investigates using formulae from physics to control Swarms and creating formations. They use virtual forces based on laws of physics concerning molecules and gravity to simpel control laws.

They investigate and prove how to make lattice formations, both hexagonal and square. They describe that hexagonal lattice formations appear when robots are placed with same distance between all, see figure1.5(a).

They later describe that square lattices can be formed by assigning two different "spin" to robots, see figure 1.5(b) where the spin is depicted as collor. Unlike spins are discance R apart while like spins are distance $\sqrt{2}R$ apart. Most of the research is based on simulations. A little of the research is an implementation on 7 real robots forming a lattice hexagon. The robots have two distance sensors mounted in opposite directions on a servo capable of turning 180°. Direction and distance measurements are quite accurate yet the swarm is very slow moving.



(a) (b)

Figure 1.5: *Lattice formations (a)Hexagonal (b)Square*

Swarm Formation Control with Potential Fields Formed by Bivariate Normal Functions [BAF$^+$]

This paper presents swarm formation using potential field generated from bivariate normal probability density functions (pdf). The swarm is commanded by controlling the distance between the members and the shape of the formation. Furthermore it provides a feature of obstacle avoidance.

First of all the formation is created by adding a repulsive and an attractive forces the robots between it is possible to maintain the same distance between the robots. Then a bivariate normal functions are used to create the vector fields that control the velocity and heading of robot swarms, with the swarm located on the surface created by the function. The potential field method does have a problem with local minima, but because the vector fields are dynamically changing the chances of hitting a local minima are much smaller.

From testresults on both simulation and implementation it is proven that the pressented approach is simple and computationally efficient compared to others. Furthermore it is able to scale well to various sizes of the swarm.

A Lightweight Formation Control Methodology for a Swarm of Non-honolomic Vehicles [Sie05]

A paper where all research is based on simulations. It proposes using a vitual leader that is connected to all vehicles via virtual "Spring laws". Individual vehicles are connected to each other in much the same way. Each virtual spring force is calculated locally. This gives the possiblity of setting and maintaining arbitrary formations. The virtual leader is placed in centre of mass and functions as swarm control. When the leader is moved alle vehicles will be moved by the force of the virtual springs. A geometric modelling of obstacles is suggested. They claim model simplifies calculation of obstacle avoidance. Communication is required to send information to all robots about the virtual leader.

Based on the benefits from the readinggroup it was decided that the scope of the project will be on formation of swarm robots. In the next chapter the provided system will be analysed through a discription.

Chapter 2

System Description

Based on the analysis of previous work, the scope of this project is narrowed down and it is now interessting to discribe and analyse whether our provided system is capable of assisting within this scope.

2.1 More Than One Laboratory

This project is a part of the groups working with small swarm robots using the More Than One Laboratory [Ngo] at the university. This section will discribe the hardware and functionalities provided by the laboratory. The laboratory has its own room at the university containing an arena useable for testing small robots. Furthermore are powersupplies, oscilloscopes, multimeters and a large variation of electrical components and mechanical spareparts for the robots. Furthermore is a stationary computer available usable as a basestation for controlling the robots aswell as datacollection.

The laboratory has a 300cm by 270cm arena with a 15cm wall around it usable to test the swarmrobots in a closed invironment. The floor of the arena is a green carpet providing an even and smooth surface with good traction.

Above the arena is a Logitech webcamera attached able to provide overview of the intire arena. The camera is connected to the basestation computer in the laboratory. By using a tracking software called Swistrack [?], it is possible to collect the possition data of the robots only using image processing as illustrated in Figure 2.1. This functionality is very usefull to analyse data, but could also be used on the fly, providing coordinates to the robots, able to react from these informations. Swistrack is a free licens and developers around the world have the possibility to program functionalities in form of filters and imageprocessing algorithms to the program.

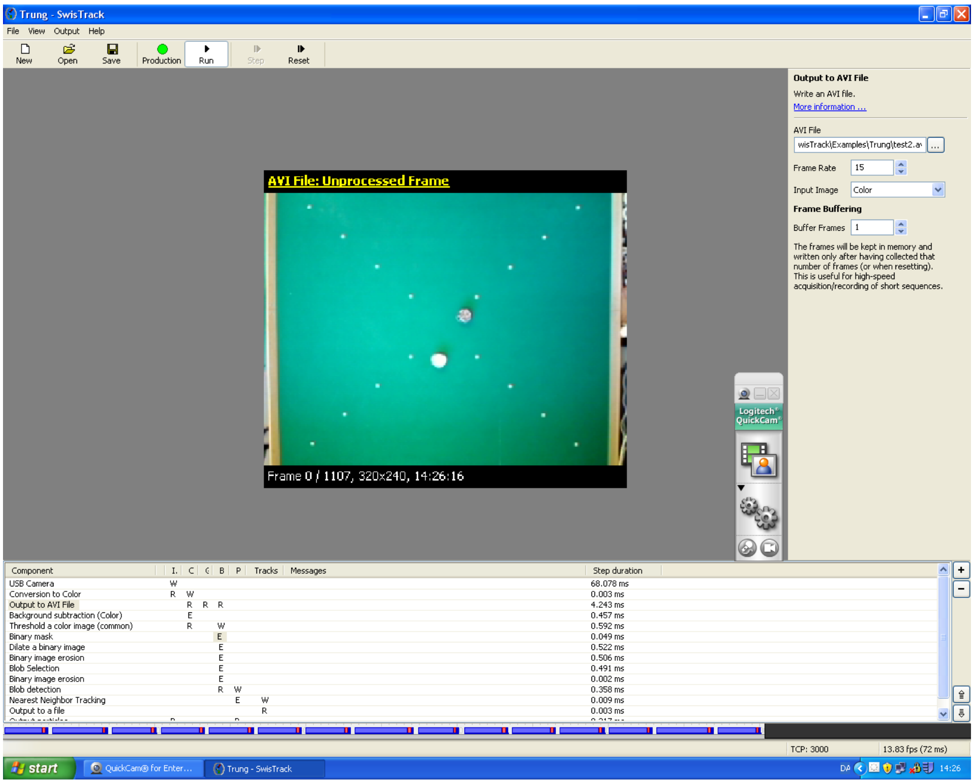
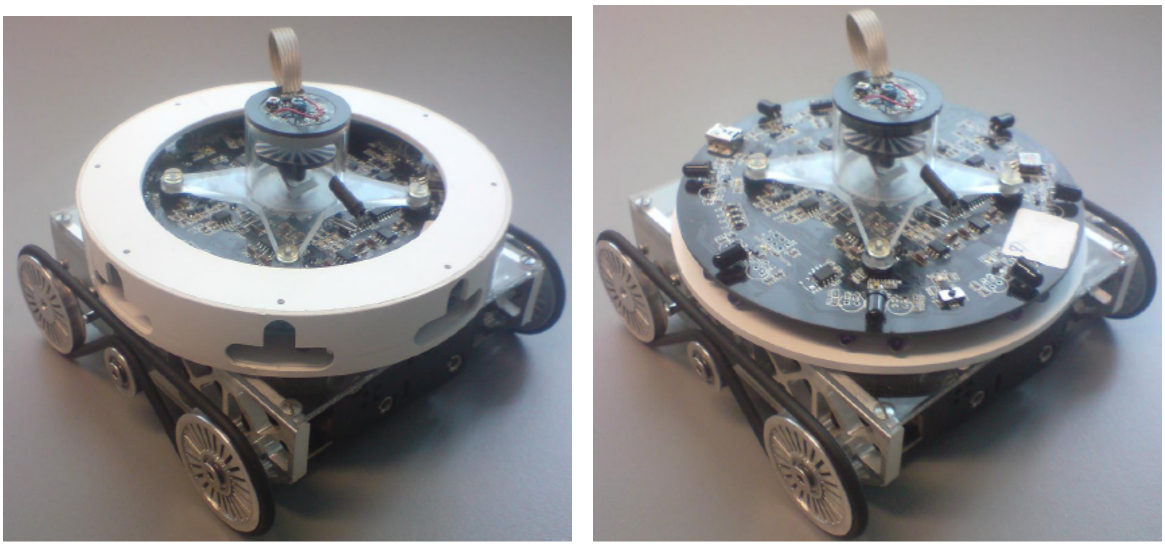


Figure 2.1: *The Swistrack GUI*

2.2 Robots

From previous projects a set of 10 small swarm robots are provided to use in this project. The following will discribe and analyse wheter the functionalities of these robots are sufficient enough within the scope of the project. The robots are illustrated in Figure 2.2 both with and without the body attached.



(a) (b)

Figure 2.2: *The provided robots with and without the body*

Mechanics and Propulsion

The robots are based on a square 13cm by 13cm chassis made of aluminum, which makes them robust and very hard to break. Two aluminum rims are attached with a rubberband around them on each side, creating a track like known from tanks. The tracks are providing good traction to the surface and makes the robots able to turn around its own axis, but they have often fall of the rims unable to proceed. Furthermore is the friction in the whole transmission large. When the robots are running around each other they have a tendency of clinging because the rims are attached outside the frame, as illustrated on Figure 2.3. The on/off button is placed outside the robot on the side of the chassis. This is a very exposed position and is occasionally switched off by other robots.

Figure 2.3: *Illustration of the problem of robots clinging*

Because of the heavy weight of the aluminum chassis and the large friction in the transmission the propulsion of the robots are created from two large 5 volt DC gear motors modelnumber HG25B from Hennkwell. The batteries used are 8 cells of recharchable AA NiMH batteries ranging between 2500 and 2700 mAh, delivering around 9.6 volts. NiMH batteries are the most commonly used cells the this time, but also some of the heaviest of its size. The technology is old and the development of new types of cells has increased the last couple of years, providing longer runtime and lighter batteries.

Boards

The robots consist of two boards. One at the bottom taking care of the motor voltage and one on the top controlling the algorithms implemented on the microcontroller.

The board at the bottom provide power to the topboard by converting the batteryvoltage to 5 volt using a regulator. Furthermore is the PWM signal from the topboard controlling the motors converted through a H-bridge to a voltage on the motors. A problem occur when the propulsion of the robot is started. The powerfull motors draws a lot of current in a peak from the batteries which have problems following this demand. This pulls down the voltage provided from the batteries resulting in too low voltage to the topboard making the microcontroller to restart. We mounted a capacitor directly on the battery, which temporary solved the problem.

The top board is the mainboard controlling the robot from a microcontroller. The microcontroller is an 8-bit ATmega128 from Atmel, providing 128 KBytes In-system programmable flash memory. Around the robot is eight infrared sensors each consisting of an ambien light sensor, two transmitters and one receiver able to communicate to other robots. The time it takes for send data to reflect on an object and return to the receiver is represented by the output of the receiver in form of current. By converting it to voltage it is possible to use the ADC input port provided by the microcontroller. This is true since the amount of light hitting a given surface decreases with a relation to the squared of the distance. On the top of the robots is mounted a rotating mirror around an infrared sensor able to scan the surroundings in almost 360 decrees, only blocked by the datacable attached. The microprocessor provides two UARTS and UART0 is dedicated for communicating on the eight infrared sensors and UART1 is splitted between using a serial USB on the Tx port and the rotating mirror sensor on the Rx port.

The robots are using the two 16 bit counters for the PWM signal to control the propulsion of the two motors. But the microcontroller also provides two 8 bit counters that are unused which are able to give a variable control of 256 steps, which is sufficient.

The microcontroller provides a Master/Slave SPI serial interface that is unused which could be used for additional functionalities. Furthermore is a Two Wire Interface (TWI) provided able to communicate with other microcontrollers, which opens the possibility of adding more boards to the robots.

Chapter 3

Test Specification

In this chapter requirements will be outlined for implementing the desire of creating a formation based on the distance between the robots. Furthermore the specification of a set of final tests are explained.

3.1 Functionality of the Robot

Initially a set of functionalities is described and later a set of requirements is attached to each functionality which will form basis of the specification of the tests performed in Chapter 5 and discussed in Chapter 7. The requirements will be expressed in paragraphs, in order to give each of them a unique number for reference. Furthermore the environments that are used for the tests are presented.

3.1.1 Functionality

The whole system is based on some overall functionalities which are based upon the scope of the project. It is therefore defined that the robots should be able to:

- Move into a possition where the wanted distance to the neighbour robots is achieved.
- Keep the achieved distance to the neighbour robots.
- Align its phase to the phase of the neighbour robots.

To make sure that the robots are fulfilling these functionalities in a satisfying way, a set of requirements are set according to the functionalities. These requirements are based on the test environment.

3.1.2 Requirements

§1. The deviation of the distance between the neighbours should be within 10cm.

This distance is chosen based on the length of the robot.

§2. The robot should be able to align its phase to the other robots within 45°.

This value is chosen based on the fact that the robots have eight sensors giving a resolution of 45°.

§3. The robots should meet the requirements for phase and distance within 15 minutes.

Based on analysis of previous work it is expected that reaching a formation within 15 minutes is a good result.

To be able to test whether the robots fulfill these requirements a set of tests are defined with the purpose of testing the functionalities of the system.

3.1.3 Environment

The tests are representing a set of scenarios which will reflect the wanted functionalities to test their requirements. The tests are performed in a simulation environment replicating the laboratory arena discribed in Section 2.1. In the following the scenarios are described according to their functionalities and their expected behaviour.

3.2 Simulation Test

The robots are placed in random initial possitions expecting to expand into a formation of equally distance to their neighbours. It is expected that the robot will do the following:

1. The robot senses its neighbours.
2. The robot moves away or towards the neighbours depending on the wanted distance.
3. The robot settles when the wanted distance is achieved.
4. The robot align its phase to the phase of the neighbours.

All requirements have now been specified and will form the basis of the design of controllers and algorithms. Furthermore they form the basis of the final test, which will be performed after the design and implementation of the controllers and algorithms in simulation, which is a subject of the following part called 3.2.

Part II
Development

Chapter 4

A unified methodology for swarm algorithm development and evolution

4.1 Abstract

Swarm robotics is a biologically inspired field which inherits significantly from the observation of biological populations such as social insects like ant colonies, termites, bees, and wasps, inspiring the vision of how a large number of simple individuals can collaborate with other members to release intelligent systems. However, it is as a complex system hard to be understood, analyzed, and synthesized and released in our daily life because of the lack of general methodology. To replicate this in the real world, an empirical methodology is required to overcome the practical challenges which arise when working with multiple robots. In this paper, a unified methodology for on the fly programming, debugging, analyzing and synthesizing based on Matlab Simulink and Real Time Workshop will be developed. The kinematics and dynamics of mobile robots as well as swarming algorithms will be virtually implemented in Simulink models which are easily modified and updated. A graphic User Interface (GUI) is built up for on the fly debugging and analyzing process with sensory data real-time updated from the robots. Instead of hand-on programming the robots, an automatic code interpreter managed by Real-time Workshop will synthesis functions modeled in Simulink in order to generate and download the code to the robots via wireless communication directly. We expect, that by using the closed loop developing model - a unified methodology - the swarm algorithms will be empirically evolved, and decentralized control of robots will be reliably released.

4.2 Introduction

Nature has many fascinating creatures which have adapted to the environment they live in through millions of years, to achieve a better chance of survive. The result is very interesting knowledge from the perspective of an engineer. The shape, physics and behavior of living creatures have been studied to replicate the functionalities with the advantage of improving human made designs. Particularly the swarming behaviors of animals like ants, wasps and birds have given huge inspiration. The usability of a swarm is larger than the sum of the individuals, which can make it more flexible, stronger, more robust and adaptable, etc. An example of this is a flock of birds forming a V-shape to reduce the air resistance and to gain a lift. Hence the individual bird is using less energy to travel a longer distance than it would do on its own. Another example is that ants stand on top of each other to create a bridge for the colony members to cross over gabs much longer than their body length. These collaborative methods have been replicated by humans for thousands of years but the engineers of today are applying them to robots for use in hazardous environments like a rescue recovery or in war zones etc. One could imagine a swarm of robots send into a crashed building searching for survived people, and by working together they can cover a larger area faster. Animals living in swarms do not have a specific leader to distribute tasks, which reminds of a decentralized system as we know it from distributed system in computers and networks. Replication of bio inspired formations into algorithms applied on robots is the main topic of our project, and they will be tested in a simulated environment as well as a real world arena. When working with swarms of robots the most important subject is the development of swarm algorithms. Due to the traditional approach, it is a time consuming task including hands-on programming, analysis, synthesis, and even evolution. For this reason a unified methodology is needed to ease such a task when developing and verifying swarm algorithms, which is the topic of this paper.

4.3 Related Work

A swarm of mobile robots like social insects can be understood as complex systems because of its properties, e.g. simple and random individual behaviors vs. robustness, adaptability, and scalability of well-organizing group. It is therefore hard to deal with such systems in formal methods. There are many different approaches to understanding, analyzing, and synthesizing the systems with an unlimited range of mathematical modeling, simulation, and empirical experiments. Mathematical modeling is a formal method of engineering scientists, mathematicians and biologists to have better understanding and prediction about the natural behaviors of social insects. Therefore, it is often used to modeling and analyzing collective actions of robot swarms. There are two tendencies of mathematical modeling named microscopic (bottom-up methodology) and macroscopic (top-down methodology) models. In microscopic models [e.g., see 3,4,5], the swarm model is built up by synthesizing instinct arbitraries of individual robots for a simulation. Although this method might provide prediction of internally sophisticated oscillation of robot swarms, it obtains many drawbacks of neither clarifying influences of external events to the swarm nor indicating the computation cost according to exponentially increased complexity. In contrast, the macroscopic model [e.g., see 2,7] is seemingly preferable because it can overcome the drawbacks of the microscopic one by abstracting the swarm behaviors at very high level while ignoring the basic instinct of individual robots. The model is easier to build up and it can take all external influences into the overall systems without computation cost. However, the model is only used to predict the general properties of the swarm while property of individuals is not traceable. Ultimately, both mathematical models can only be used to get better understanding about the swarm behaviors, it is not clear on how to transfer the simulation results of the models to the real world. Another methodology of swarm algorithm development is based on simulation evolution [e.g., see 6]. Evolutionary algorithms in association with neutral networks are usually used to improve the fitness function by selecting the best candidates in each generation for the next generation evolution. The overall behaviors of robots are evolved and trained through a computer simulation. It is then synthesized in order to generate the decentralized control for real robots. Although this method has high potential in reducing the computation cost and time consuming in swarm algorithm development, it still has the significant problem of transferring the simulated code to the real robot due to the big gap between the simulation environment and the real world, which is not easy to fill up. A practical methodology to the swarm algorithm development is based on experiments of real robots [e.g., see 8,9]. The swarm algorithm developed relies on observation and data collection of experiments that makes a true sense about the swarming behaviors in the real world. However, it is not easy to transfer the knowledge synthesized from the observation and data analysis to the swarm algorithms and decentralized control of robots. It is a hard and expensive job due to doing hand-on coding for many robots with heterogeneous bodies. Although existing approaches contributes to the step of understanding the complexity of robot swarms as well as development of swarm algorithms and decentralized control, none of those are realistic approaches to the swarm algorithm development and evolution.

4.4 Methods

This paper proposes a new methodology for developing and testing swarming and formation algorithms. This methodology incorporates many of the good features from the methodologies mentioned in Related Work. It is focused on bridging the gaps between simulation and real the world as well as increasing understanding of swarms by combining the different approaches and streamlining analysis.

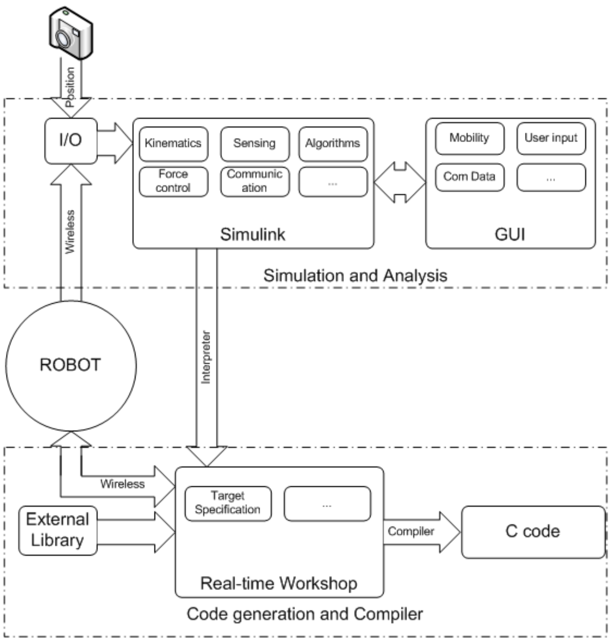


Figure 4.1: *Diagram of the proposed methodology*

The proposed methodology involves three main function blocks working in a closed loop for development and evolution: Matlab Simulink, GUI, and Real-time Workshop. On one hand, swarm algorithm can be internally simulated through mathematical models built in Simulink, and shown in GUI for pre-analysis. On the other hand, the sensory data can be directly imported in the same model for the empirical verification. Analysis of data from experiment and/or simulation are presented in a high level GUI making it increasingly easier to compare, and therefore transfer results from mathematical models to the real world through the Real-time Workshop. The Real-time Workshop plays a role as an interpreter that translates model-based algorithm and control to C code before compiling to machine code and downloading to the robots. The wireless connections to and from the robots are imperative to the proposed methodology for two reasons. Firstly sensory data is needed in real-time to be imported into the Simulink for processing and analysis. Secondly in order to automate the expensive job of hands-on-coding of robots, the machine code can be remotely downloaded to the robots. To achieve full hands-off coding of robots and keep focus on the model-based swarm algorithms, the GUI allows to change parameters on the fly. A diagram of the methodology is shown in Figure 4.1.

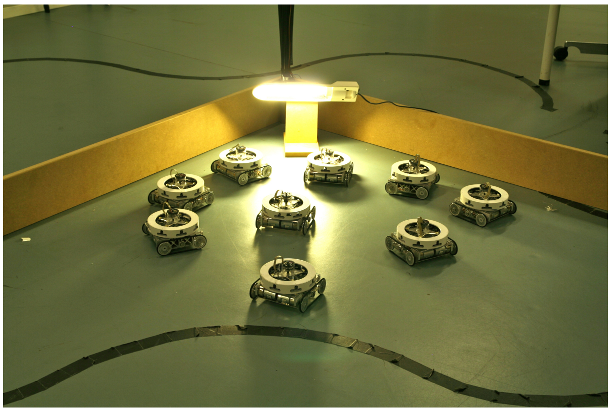


Figure 4.2: *Provided robots in the arena*

4.4.1 Development

A set of 10-15 small simple robots is provided to us by the university, from a previous project as illustrated on Figure 4.2. These 13 by 13 cm robots are driven by a tank drive train powered by two small electrical motors. The drive train gives the individual robots great moving ability since they are able to turn around on their own center axis. Hence it is able to change position fast with minimal use of movement. Furthermore they are able to change the drive train between the caterpillar tracks and a four wheel drive, depended on how the tracks are attached. This gives advantages in rough terrains. The main circuit-board of the robots is provided with an ATmega128 microprocessor which is programmed trough an In System Programming port. The microprocessor can however be programmed using any of its communication ports. Furthermore is a set of 8 infrared distance sensors attached around the sides, and one on the top with a rotating mirror, able to scan the surroundings. The robots come with a 300 cm x 270 cm arena with 15 cm high side walls ensuring that no robot can escape. The arena is covered with felt carpet which gives good traction.

The kinematic model of the robots is implemented in the Simulink Model for simulations. In this way the output from simulation and experiments will be directly comparable. Also A communication model is developed for receiving and sending data from the PC's wireless communication port into the Simulink model. Simulation of the swarm will give values and formation to compare to the real world measurements. On top of the Simulink model a GUI is designed to show all sensor data, as well as the formation of the robots. Furthermore it adds ability for a user to change the parameters of the algorithms on the fly for faster tuning of these. By using Real Time Workshop combined with a GCC compiler the code generation from the algorithms input to the GUI is automated. This and the ability to program the robots through a wireless connection, as demanded in the methods, automate many time consuming jobs. The existing robots will be expanded with wireless radio communication. To implement the wireless link a new Printed Circuit Board (PCB) will be made for the robots. The PCB will be designed using premade radio modules. This removes the need to design an antenna and communication stack. An ATMega128 Microprocessor will be used on the board. This is necessary since the tasks for the new PCB will be to handle all radio communication, to control the motors and wheel encoders and to detect if there is floor beneath (cliff sensors). All of this cannot be implemented on the existing board since there are not enough serial UARTs, and ports left. The only available communication port is a two wire serial interface. Therefore the new PCB will interface with the Old board using I^2C. It would be desirable to have a radio with low power consumption for sensor data and a high bandwidth for programming. Since high bandwidth and low power consumption are conflicting requirements two different radios will be used.

The existing arena gives a well defined environment. However consistency in measurements still demands a proper sensor setup. A pseudo-Global Positioning System (pGPS) in the form of a camera fixed

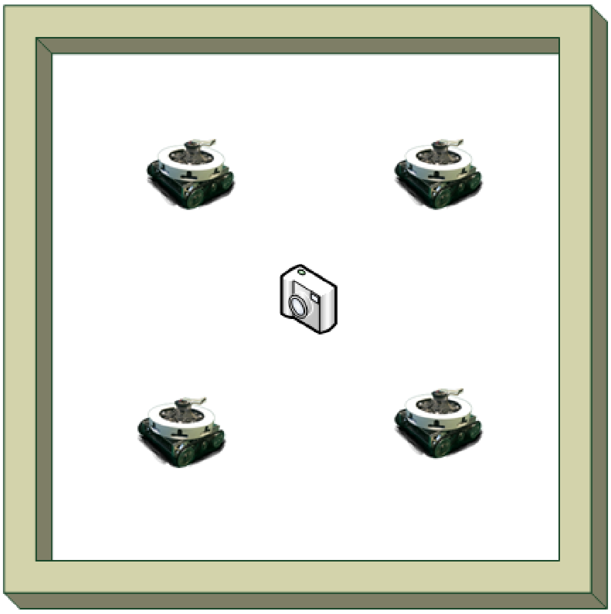


Figure 4.3: *Arena and pGPS camera*

above the arena is implemented as shown on Figure 4.3. The camera will be used for visual as well as position documentation. The pGPS data can be used as input to algorithms on the robots as well. By implementing the pGPS in the arena the verification system is not dependent on the robots. This can be an advantage since data delay and processing is not affected by algorithm implementation. The precision of pGPS depends on the resolution of the camera compared to the arena size. Software is used to identify robots and track their movement in the picture frame. The picture frame is distorted since the distance from the camera lens to all points on the arena floor is not the same. Software to overcome this is available as open source. [1]

4.5 Results

At the time of writing the development is still a work in progress based on the methodology. However the methodology is set and the development is based on this. The results and conclusion are based solely on the methodology. The analyses of the swarms are eased by the GUI leading to better understanding. Since the GUI is the only needed interface to the system focus can be held solely on algorithms. Real time comparison of sensor data and simulation. The task of programming the robots is simplified significantly.

4.6 Discussion and preliminary conclusion

This paper is to illustrate the methodology of a larger project. In this paper, the development of a productive methodology has been in focus. It is of most importance when working with a large number of robots that it is easy and fast to try out developed algorithms and non the least to improve on them. The methodology proposed in this paper is model-based, which ease understanding and modification in development pipeline. It is believed that this will save time in the evolution of robot swarms A wireless communication module for the robots is produced and connected to a Matlab Simulink model and GUI. On one side, it makes the uploading and changing of parameters in algorithms fast and easy, without having to plug and unplug the individuals to program them. On the other side, it allows to download the code to several robots or to control them simultaneously. More specifically, the wireless link opens possibilities for many options which can be operated real time. We are able to read out sensor data, which can save a lot of time in debugging. We can determine whether the robots should be able to communicate with others that are not direct neighbors. The parameters of the algorithms can be tuned or optimized automatically real time, through a real world experiment. It is achieved by using measurements from sensors, to use either in a simulation or in the experiment itself. By adding a camera above the arena, we are able to locate the robots position and orientation which can be used as a factor when developing the swarm algorithms. It makes it easier to achieve the correct positions. Furthermore it is important that the algorithms can be tested, meaning that the robots are doing what they are supposed to do. The position is useful to test whether the result is accepted or not. The use of known standards in components is believed to ease the development of the hardware as well as the software. They are all well documented and hold a lot of examples. In time of writing the test setup is not completely finished, but the communication hardware is designed and ready for production, and the software combined with the Simulink model is designed as well. An analysis on some of the work, which is related to this project, is made. We have taken a look into the state of the art projects which have given us a great insight in the possibilities and limitations of swarm robots. Now the overall methodology and the progress of the development has been outlined. Next the rest of the work will be discussed, and will mainly focus on the development of algorithms.

4.6.1 Future Work

The hardware and software based on the methodology presented in this paper should be finished next. When it is done the work can begin on the applications it is designed for; namely swarm algorithms. Ideally when this setup is up and running the shift from one algorithm to another is just a click in the Matlab GUI. To outline our thoughts on the rest of the project, some of the future perspectives that could be interesting to look at are described in the following. Develop swarm algorithms capable of tuning their own parameters runtime. Algorithms that adjust its parameters to fit the present environment would be an interesting step in the evolution of swarm robotics. Test heterogeneous algorithms i.e. not all robots having the same algorithm. Just like ants in nature have different jobs as guards, food finders or food carriers. Challenges would be to get robots with different algorithms to collaborate on the same task. Adapt the setup, to work via Internet making a remote lab accessible for anyone around the world that needs to test an algorithm. Considerations about battery time cost and local assistance would be some of the big challenges.

4.7 References

[1] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, and A. Martinoli. SwisTrack - A Flexible Open Source Tracking Software for Multi-Agent Systems. In Proceedings of the IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems (IROS 2008), pages 4004-4010. IEEE, 2008. [2] O.Soydal and E. Sahin. A macroscope model for self-organized aggregation in swarm robotic systems. In E.Shain, W.M.Spears and A.F.T.Winfield, editors, Swarm Robotics- Second SAB 2006 International Workshop, volume 4433 of LNSC, pages 27-42, Berlin, Germany, 2007, Springer-Verlag. [3] Tamas Vicsek, Andras Czirok, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel type of phase transition in a system of self-drived particles. Physical Review Latters, 6(75): 1226-1229, August 1995. [4] Tad Hogg, Coordinating microscopic robots in viscous fluids. Autonomous Agents and Multi-Agent Systems, 14 (3): 271-305, June 2006. [5] A.Galstyan. T.Hogg, and K.Lerman. Modelling and mathematical analysis of swarms of microscopic robots. In Proceedings of IEEE Swarm Intelligence Symposium (SIS-2005), Pasedena, CA, pages 201-208, Los Alamitos, CA, June 2005, IEEE press. [6] V.Trianni, Evolutionary

swarm robotics - Evolving self-organizing behaviours in groups of autonomous robots. Vol. 108 of Studies in Computational Intelligence, Springer-Verlag, Berlin, Germany, 2008 [7] A.Martinoli. Swarm Intelligence in autonomous collective robotics: from tools to the analysis and synthesis of distributed control strategies. PhD thesis, Ecole Polytechniques Federale de Lausanne, 1999. [8] D. Payton, M. Daily, R.Estowski, M.Howard, and C.Lee. Pheromone robotics. Autonomous robots, 11(3): 319-324. Nov. 2001 [9] F.Mondada, L.M. Gambardella, D. Floreano, S. Nolfi, J.L.Deneubourg, and M.Dorigo. The cooperation of swarm-bots: physical intefaction in collective robotics. IEEE Robotics & Automation maganize, 12 (2): 21-28, June 2005.

Chapter 5

Simulation of Formation Algorithms

5.1 Abstract

This paper is part of a larger project on swarm robotics. The goal is to mimic the simplicity and redundancy seen in biological swarms to a robotic swarm. The field of swarm robotic is inspired by biology such as a swarm of insects, a flock of birds or a school of fish. The vision is that a large number of simple individuals can collaborate to release intelligent systems and capabilities far greater than one individual. In robotics it translates to having a swarm of simple robots achieve complex tasks. Some of the envisioned benefits from these swarms are adaptability, cheapness, redundancy and robustness. Potential fields are one of the preferred control methods for Robot Swarms. Similar to nature, swarming has been achieved by others using a combination of potential fields, obstacle avoidance, directional alignment and speed matching. So far the implementations on real robots have had reliable 360 degree sensing and reliable intercommunication. All animals however have limited sensing capabilities and communication is somewhat unreliable. Reliable sensing and communication is expensive on a robot and it strives against the wish to make the robots as cheap and simple as possible. This paper proposes modeling sensing and communication limitations into robot swarm simulations while maintaining the simple control of potential fields. The limitations in the modeled robots are blind spots in the directional sensing and unreliable intercommunication. The limitations are based on an existing robot design on which the results will be implemented in another paper. Since the robots are all alike it is a homogeneous swarm. The robots are circular and differential driven and each robot have 8 directional IR transmitters placed with 45 degree spacing. An IR receiver is placed just below each transmitter so reflections from obstacles can be detected as well as signals from other robots. The robots are modeled in Matlab Simulink where the communication and sensing capabilities are described using probability functions. The swarm is simulated in Matlab using copies of the robot model running in parallel which mimics the control being calculated locally by each robot in the swarm. The control is implemented as a virtual potential function, a directional alignment or Phase controller and an obstacle avoidance controller. To control the whole swarm an external force can be set to attract or repel the robots. We have proven that it is possible to maintain swarm connectivity, obtain and maintain formations based on potential field. We have also shown that a phase alignment control not necessarily improves formation stability.

5.2 Introduction

The behavior of a swarm can be hard to describe since it is depending on the task it performs. But the goal of this project is to get them working in a sort of formation, whether it have strong or weak connectivity. This behavior is interesting because, if it is possible to maintain a formation and create a model of this, it is possible to control the whole swarm instead of the individuals, without any designated leader. What is wanted is a fully distributed swarming algorithm in the sense that each robot determines its reactions based on its own sensory data.

This paper works with three different levels of order in robotic swarms. Each level is increasingly more difficult to maintain. The most basic is obstacle avoidance. If a robot cannot identify obstacles and avoid them, the robot is not stable. Next level is connectivity. If connectivity is not maintained in some level a swarm is not stable. The highest level is formation. To maintain a formation in a swarm requires: obstacle avoidance, connectivity and alignment of inter robot distances and orientation(phase). Based on the related work it is possible to consider the positive and negative experiences, and act from that.

5.3 Related Work

To maintain coherence, an artificial potential field is one of the preferred control methods in mobile robotics. However, most of potential field is based on the assumption of perfect sensing capability in detecting the neighbours while distinguishing between robots and obstacles [TCGS08]. This is not a realistic assumption in biology or engineering because animals have limited sensing capabilities and unreliable communication, and so do nowadays robotic technologies.

It is stated in [SSHH02] that it is sufficient for each robot to know distance and angle to its neighburs to achive and maintain a hexagonal lattices formation. The angle measurement on the robots developed for this project however is discrete in intervals af 45°and the distance measured as a light intensity level will vary from 0 to 1.5 meters within one such angle step even though the light level is the same. It is therefore not given that it is possible to maintain a stable lattices formation.

The distance based potential field, as used in[SSHH02], is good at keeping a constant distance between robots using repulsive and attractive forces, but have problems aligning the robots towards a common phase. That problem is solved by [TCGS08] using communication, because the robots then can share their relative phase with each other. This paper presents a method to maintain a swarm formation using both distance measurements and line of sight communication as input of potential field forces to gain reliable formation connectivity and alignment.

5.4 Method

Control of the robot swarm is described in equation 5.1 where based on its observation and interaction, robot r calculates a force controlling its movement.

$$\vec{F_r} = \sum_{i=1}^{8} \vec{F_{pot}}(i) + \sum_{i=1}^{8} \vec{F_{phase}}(i) + \sum_{o=1}^{8} \vec{F_{obst}}(o) + \sum_{g \in G} \vec{F_{SWARM}}(G) \tag{5.1}$$

$\vec{F_{pot}}$ is a probabilistic virtual potential field force between robots based on communication and measurement of light intensity. Combination of the internal attraction and repulsion induces coherence in the swarm formation with accordance to both distance and orientation. $\vec{F_{phase}}$ is a phase alignment force between robots based on directional encoded communication, $\vec{F_{obst}}$ is a probabilistic repulsive force between robots and obstacles. This repulsion ensures all robots to avoid obstacles when swarming.

The fourth item $\vec{F_{SWARM}}$ is a virtual force input to control the entire swarm. It illustrates the goal applied to the robot r, to achieve the common goal of the swarm. The swarm formation is based on the probability of limited communication and sensing of individual robots, which allows designing a fully distributed control for swarm formation. The advantage of the probabilistic potential function is that it is not dependent on the environmental condition as the probability of communication is equally changed for all robots.

Note that $\vec{F}_{obst}$ is not implemented in this paper since no obstacles will be introduced in the simulation. The common goal $\vec{F_{SWARM}}$ of the swarm is simply set to zero. This means a robots behavior is determined as a rotational force from a phase alignment controller added to a directional force from a potential field controller:

$$\vec{F}_r = \sum_{i=1}^{8} \vec{F_{pot}}(i) + \sum_{i=1}^{8} \vec{F_{phase}}(i) \tag{5.2}$$

The 8 directional forces in equation 5.2 have fixed angles equal to the robots 8 fixed sensors as seen on figure 5.1. The summation in equation 5.2 is depicted in figure 5.2.

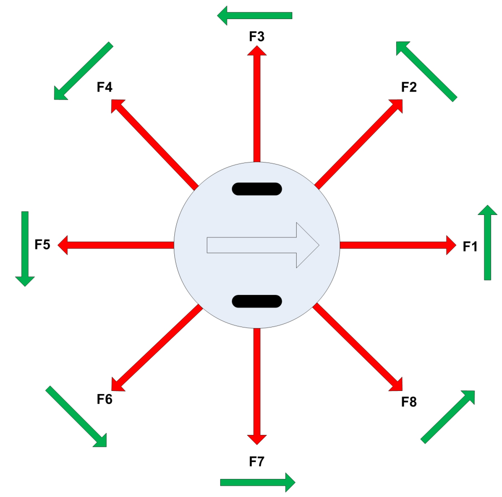


Figure 5.1: *Robot with the 8 directional forces depicted*

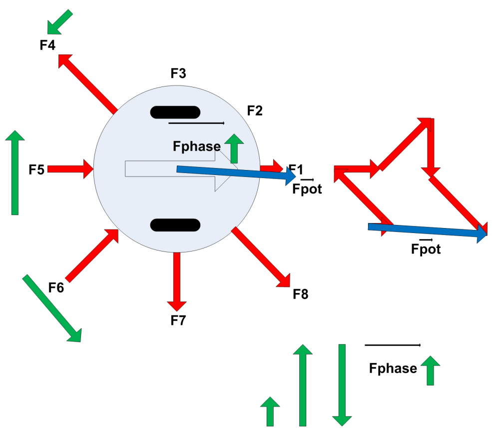


Figure 5.2: *Summation of directional forces on a robot*

The virtual forces are calculated using three controllers; an Attractive Controller, a Repulsive Controller and a Phase Alignment Controller. The reason for splitting up attractive and repulsive is clear from the description of the controllers.

5.4.1 Attractive Controller

The robots in a swarm need to be attracted to each other in order to maintain connectivity. It is important that the robots are able to distinguish between obstacles and other robots in order to avoid the swarm "sticking" to obstacles. To accomplish this IR communication is used. Whenever a robot receives a byte from another robot, a virtual force of constant magnitude *atf* is at set in the direction the receiving sensor points. In order for a robot to know that it is another robot transmitting and not a reflection each robot must transmit different bytes containing the robots ID. Since IR communication is unreliable a proper model for the communication channel must be used in the simulation. The probability of biterrors will vary with different scenarios. Therefore the magnitude *atf* should be adapted to each scenario to ensure connectivity.

$$\vec{F}_{atf} = \begin{cases} atf & \text{if byte recieved} \\ 0 & \text{if byte not recieved} \end{cases} \tag{5.3}$$

where *atf* is set to 3.

5.4.2 Repulsive Controller

To maintain a constant distance between robots a repulsive force is needed. The wanted distance between robots will be maintained when the repulsive controller and the attractive controller cancel each other out. Robots will be repelled by measuring the strength of IR light recived on each sensor and calculating a repulsive force as linear proportional to this using equation. The measurements of IR light strength are not as unreliable as communication. However since the robots transmitters are directional and robot orientation is probabilistic a model taking this into account must be used.

The repulsive force is calculated as a function of the voltage over the receiving diode U_r.

$$\vec{F}_{rep} = U_r \cdot ref + L_0 \tag{5.4}$$

where *rep* is negative in the order of -40 and L_0 is 50.

5.4.3 Phase Alignment Controller

By coding the ID byte sent from a robot with information about which transmitter number it was send from it is possible for the robots to know their relative phase difference. The phase differences for all sensors are summed and the phase error P_{error} is found. The controller is inspired by [TCGS08]

$$\vec{F}_{Phase} = P_{error} \cdot K_{phase} \tag{5.5}$$

where $K_{phase} = 0.001$

5.4.4 Modelling

To verify the coherence and formation of the Probabilistic Potential Field based swarm, a simulation in Matlab Simulink is constructed. 10 individual robots are implemented as separate subsystems interacting through the simulation platform. Output from each robot to the world is its coordinates and orientation (x, y, θ), which are calculated based on probability of communication and perception of a center receiver and 8 pairs of infrared transceivers spaced 45°. Communication probability of a robot is estimated on line-of-sight communication frequency and relative positioning when communicating with its neighbours while its perception probability is based on reflective intensity of signals broadcasted by itself.

There are many ways to describe an object or behavior and the way engineering scientists are doing this is trough a Mathematical model. This helps us to understand the object whether it is a bird or a robot. When working with robots we can use these mathematical models to both predict and control its actions based on a set of information given by a controller.

Kinematic Model

A kinematic model of the robot is a mathematical description of the capabilities and dependencies of the robot. This is needed to give a correct image to be able to control the individual robots as well as the entire swarm. The inertia is neglected in this model.

To get the right response in the simulation of the system, when applying a controller we must model the system as close to the real world as possible. Hence a kinematic model of the robots is created describing the mathematical relations between the input and output of the system. The robots are differential driven by a set of two motors controlling the right and left wheel respectively. Hence the input of this model should be a force applied to both motors. This force can be both negative and positive according to the movement wanted. If both forces are equally positive or negative the robot will drive straight forward or backwards. If the forces are unequally distributed the robot will turn forwards the side of the weakest force applied. The robot is able to turn on its own axis without any movement in xy plane by applying equally large velocities, but in opposite direction, on the motors. Figure 5.3 shows the the robot in the xy plane.

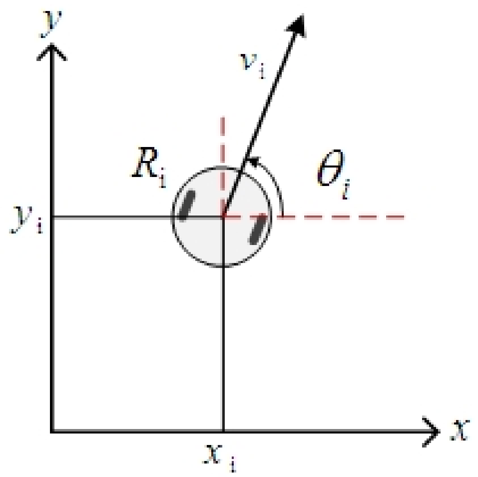


Figure 5.3: *The robot model in xy plane*

The model should be able to convert this input to an output expressed by polar coordinates with a linear and angular velocity because the relation between the forces and the Cartesian position of the robot can be expressed through a set of kinematic equations as in formula:

$\dot{X}_i = \hat{v}_{i_x} \equiv v_i \cdot cos(\theta_i)$

$\dot{Y}_i = \hat{v}_{i_y} \equiv v_i \cdot sin(\theta_i)$

where $\hat{v}_{i_y}$ and $\hat{v}_{i_x}$ are the velocity of v_i along the X and Y axises. The angle θ is the derivative of the angular velocity of the robot ω_i:

$\dot{\theta} = \omega_i$

The linear velocity v can be calculated as the mean of the forces applied on the wheels: $v_{i_y} = \frac{v_{r_u}+v_{l_u}}{2}$

Communication Model

As mentioned earlier the infrared communication is not that reliable and is far from free of bit errors. To achieve a realistic illustration of how well the communication can be used to control the swarm, a probabilistic model of the communication is introduced to the simulation. The communication model is based on the work of another student group [LSJK09] at Aalborg University. This communication model describes the error rate of sending a bit between two robots. The bit error rate depends on the strength of the signal received, which depends on both the angle and the distance between the robots. When the transmitter and receiver are pointing directly towards each other the signal is strongest and if either the transmitter or receiver robot is turned away from the other robot the signal will decrease. The amount of light hitting a given surface decreases with a relation to the squared of the distance. Furthermore the strength of the signal is also dependent on the reflections of the physical surroundings of the robots as well as their design. The tolerance of the components is also an unknown factor and the model was therefore based on a set of measurements from all 8 different infrared transmitters on the robots. Based on these experiment data a probabilistic function was made to get the relation between distance and angle relative to the voltage in the receiver. Figure 5.4 is an illustration of the signal spectrum relative to the probabilistic characteristic of the communication model, showing the error rate when sending a zero and a one bit. The model only provides a value for the probability and does not distinguish between the robot being either far away or the angle very big, hence the value could be the same. It is necessary to know the X and Y coordinates of the neighbour robot, and add it as an input to the model. This way we are able to add a probabilistic constant to our system if we know these coordinates. $P0$ in equation 5.6 is the probability of a bit error when sending a "zero", $P1$ in in equation 5.7 the probability of bit error when sending a "one", U is the voltage level on the reciving diode as a function of angle (a) and distance (d) and U_r is that same voltage with maximum limit of 4.7 Volt. The bit error is changed into a byte error using 50% ones and 50% zeros in byte.

$$P0(U_r) = \begin{cases} 0 & \text{if } U_r < 1.3640 \\ \alpha_0 \cdot U_r + \beta_0 & \text{if } U_r \geq 1.3640 \end{cases} \tag{5.6}$$

$$P1(U_r) = \begin{cases} 1 & \text{if } U_r < 0.0549 \\ \alpha_1 \cdot U_r + \beta_1 & \text{if } 0.0549 \leq U_r < 0.2976 \\ 0 & \text{if } U_r \geq 0.2976 \end{cases} \tag{5.7}$$

$$U_r(a,d) = \begin{cases} U(a,d) & \text{if } U(a,d) < 4.7 \\ 4.7 & \text{if } U(a,d) \geq 4.7 \end{cases} \tag{5.8}$$

$$U(a,d) = M_r \frac{K(a)}{d^{n(a)}} \tag{5.9}$$

$$K(a) = \frac{c_K}{\sqrt{2\pi} \cdot \sigma_K} e^{\frac{-a^2}{2 \cdot \sigma_K^2}} \tag{5.10}$$

$$n(a) = \frac{c_N}{\sqrt{2\pi} \cdot \sigma_N} e^{\frac{-a^2}{2 \cdot \sigma_N^2}} \tag{5.11}$$

Where:
$\alpha_0 = 6.9 \cdot 10^{-3}$, $\beta_0 = -9.4 \cdot 10^{-3}$, $\alpha_1 = -4.1202$, $\beta_1 = 1.226$, $M_r = 0.5271$, $c_K = 518.71 \cdot 10^3$, $\sigma_K = 6.336$, $c_N = 163.106$ and $\sigma_N = 25.768$.

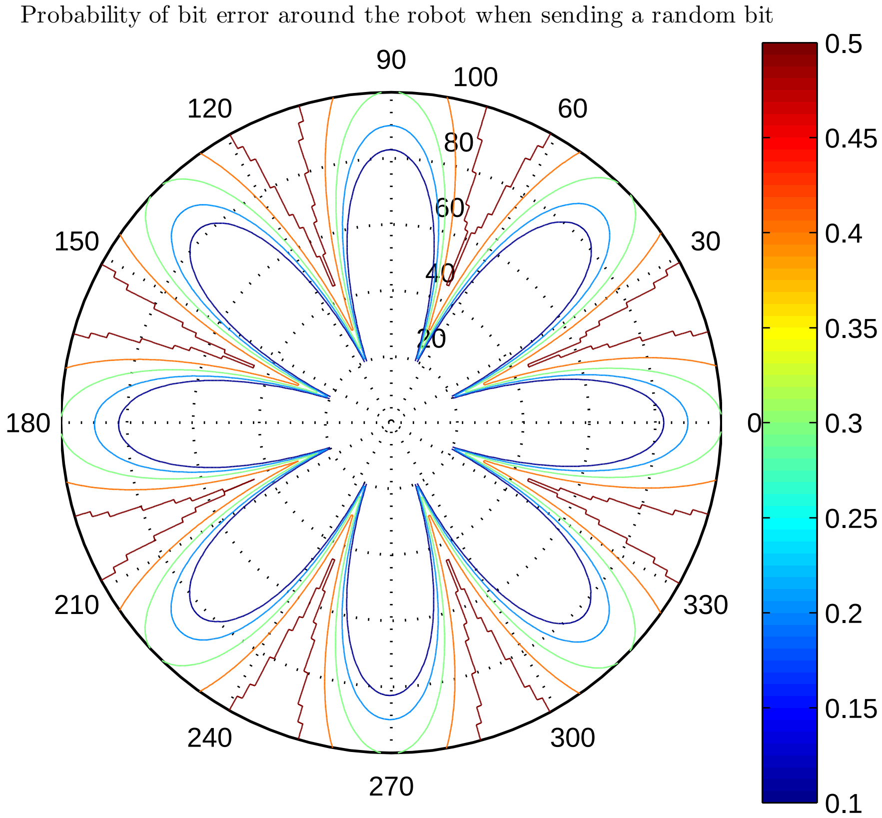


Figure 5.4: *Bit error probabilistics of a random bit. Figure from [LSJK09]*

Simulator

The simulator is built in Matlab Simulink(TM) using the kinematic model of the robots and the probabilistic communication model. Since Simulink is a graphically based each robot is implemented as a separate block as shown in figure 5.5. The robot block contain the kinematic model and controllers. Input to the robot blocks are sensor values and output are (x, y, θ). A seperate block for each robot models the environment. This block takes all other robots output and compare them to its robots output. From these informations the sensor values for the robot is calculated making sure that robots cant transmit through each other. No boundaries are implemented in the simulator as of yet. Therefore obstacle avoidance is not possible.

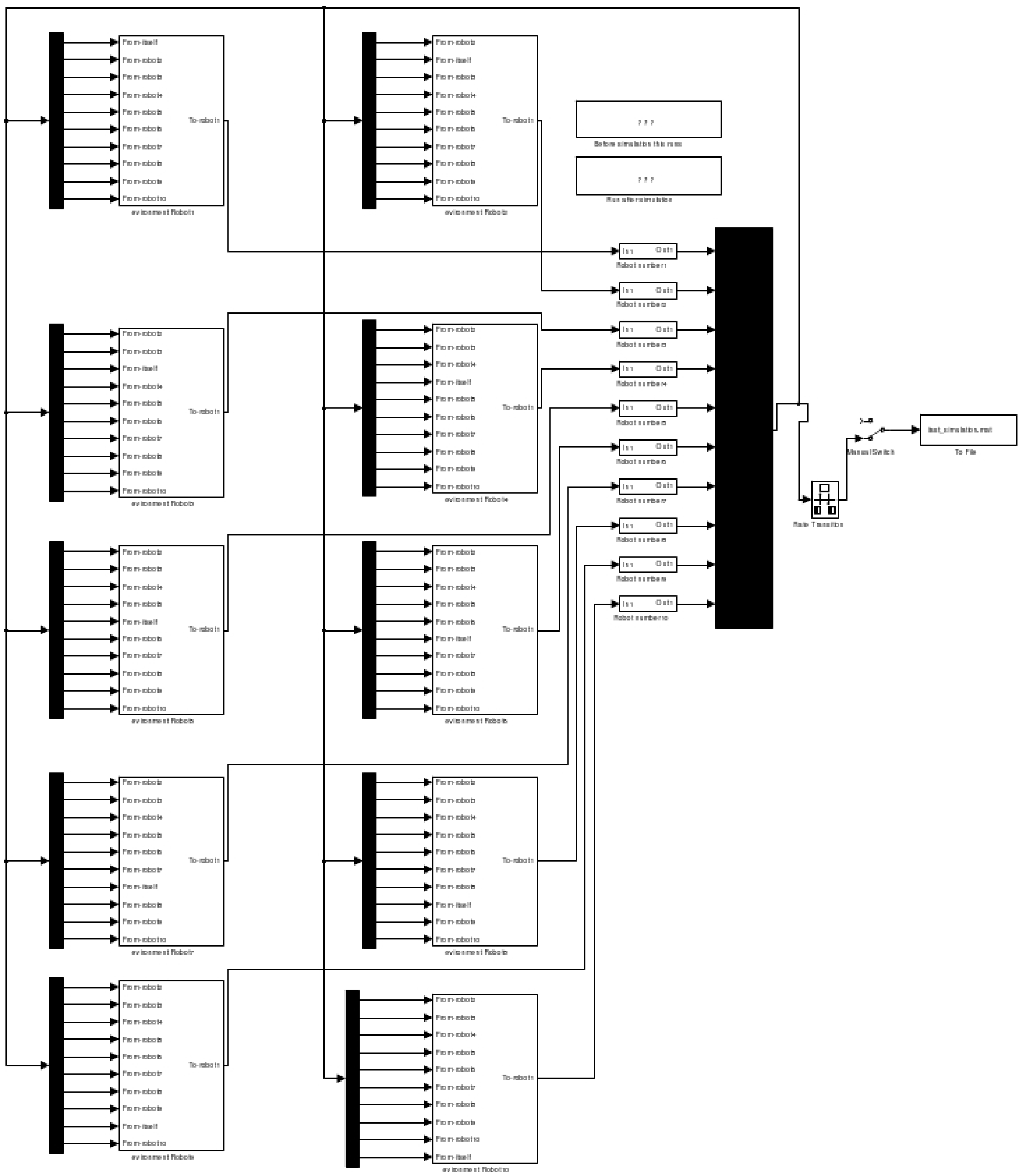
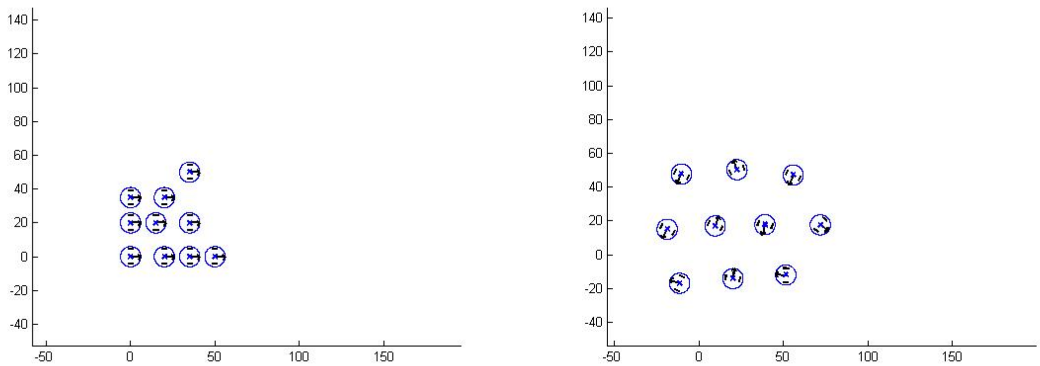


Figure 5.5: *Overview of the Simulink Simulator*

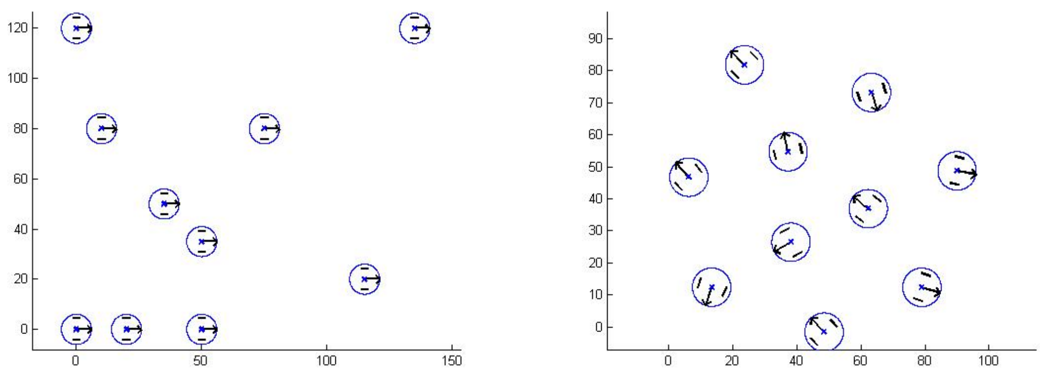
5.5 Results

Three different initial setups have been tested using potential function control. The setup 3 is then tested using potential function control and phase alignment control. Each simulation was 15 minutes. Data is plotted for every 10th second. Figure 5.6(a) shows the inital positions in setup 1, and figure 5.6(b) the end positions. Likewise figure 5.7 shows setup 2. Setup 3 is show in one minute intervals in figures 5.8 and 5.9.



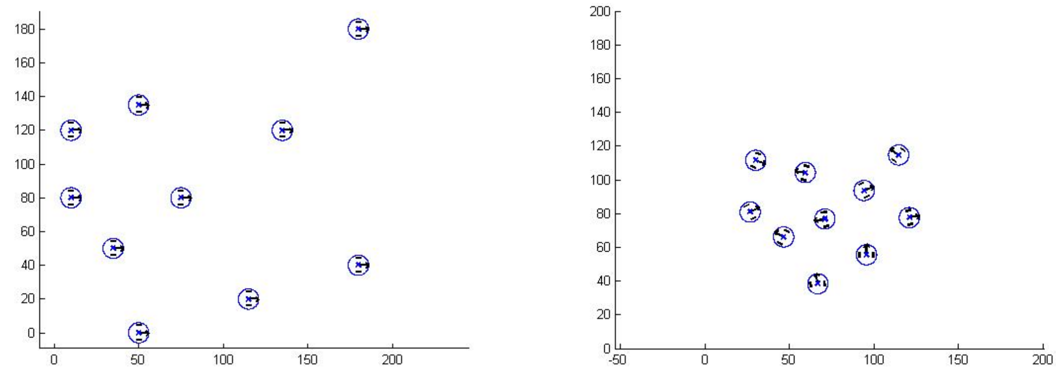
(a) (b)

Figure 5.6: *Initial setup 1:(a) Position at time = 0 s (b) Position at time = 900 s. Potential function controller only*



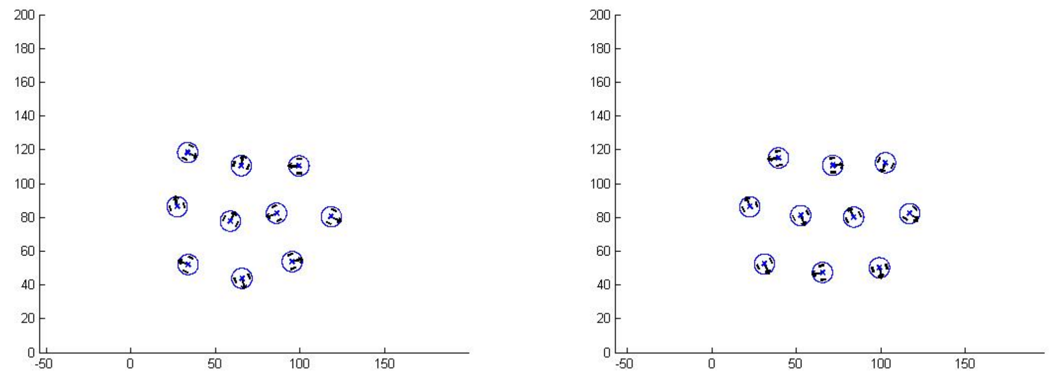
(a) (b)

Figure 5.7: *Initial setup 2: (a) Position at time = 0 s (b) Position at time = 900 s. Potential function controller only*



(a) (b)

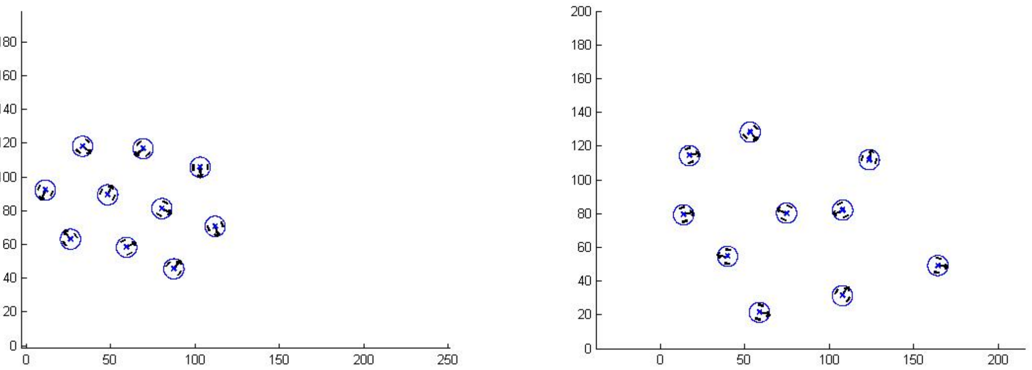
Figure 5.8: *Initial setup 3:(a) Position at time = 0 s (b) Position at time = 60 s. Potential function controller only*



(a) (b)

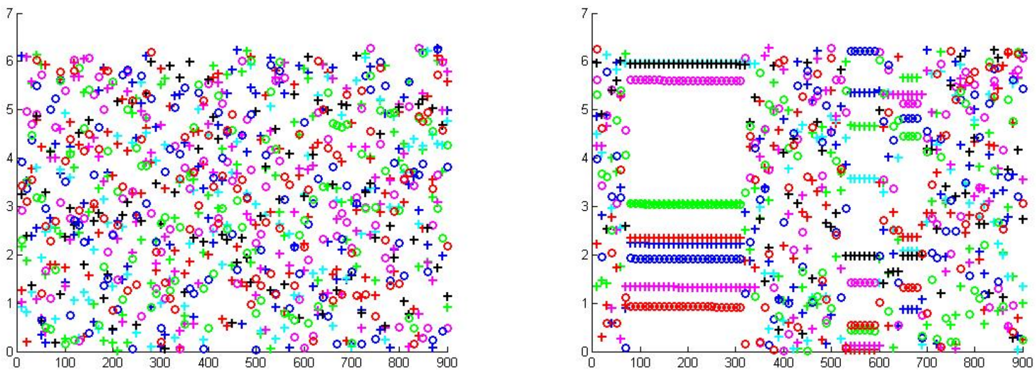
Figure 5.9: *Initial setup 3:(a) Position at time = 180 s (b) Position at time = 240 s. Potential function controller only*

Figure 5.10(a) shows the end positions for setup 3 when using potential function control and 5.10(b) shows the end positions for setup 3 when using potential function control and phase allingment control. The phase of all robots for both runs of setup 3 is plotted in figure 5.11.



(a) (b)

Figure 5.10: *Initial setup 3:(a) Position at time = 900 s (b) Position at time = 900 s. Potential function controller and phase allingment controller*



(a) (b)

Figure 5.11: *Initial setup 3:(a) Phase of all robots plotted as a function of time, Potential function controller only (b) Phase of all robots plotted as a function of time, potential function controller and phase alignment controller*

5.6 Analysis

The data from the Results section are analysed further. Figures 5.6(b), 5.7(b) and 5.10(a) clearly show equidistant lattice formations forming. It is interesting to compare the one minute plots of setup 3 (figures 5.8, and 5.9) where the hexagonal form appears after only 3 to 4 minutes to the plot of setup 2 (figure 5.7(b)) where at 15 minutes there still are no hexagonal forms. This shows a very large variation in the formation settling time depending only on the initial positions.

In [TCGS08] results show that phase alignment and proximity control combined gives a more compact lattice formation than just proximity control. That is not the case with this setup which is clearly seen when comparing figure 5.10(a) to 5.10(b).

The interrobot distances of setup 3 using potential controller are plotted as a function of time in figure. 5.12. It clearly shows that groupings of equidistant robots occurs at approx 35 cm, approx 65 cm and approx 85 cm. This corresponds very well with the interrobot distances of setup 1 in figure 5.13(a) of approx 35 cm, approx 65 cm and approx 95 cm. The forming of equidistance in setup 2 is slower as show in figure 5.13(b), but it is beginning. In comparison the interrobot distances of setup 3 using potential controller and phase controller is plotted in figure 5.14. No clear grupping of distances is there.

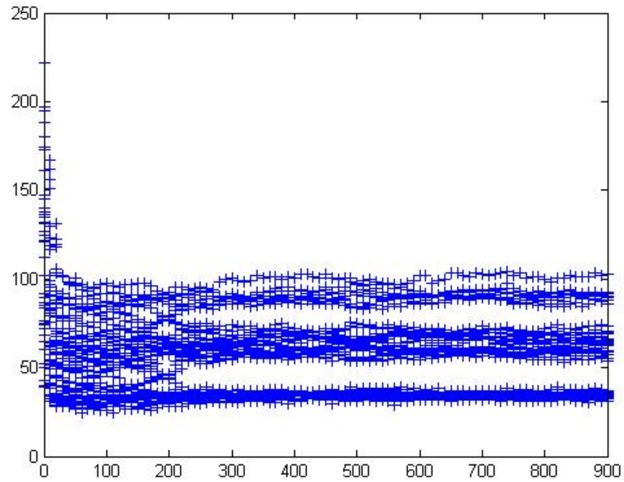


Figure 5.12: *Interrobot distance as a function of time. Setup 3 with potential controller*

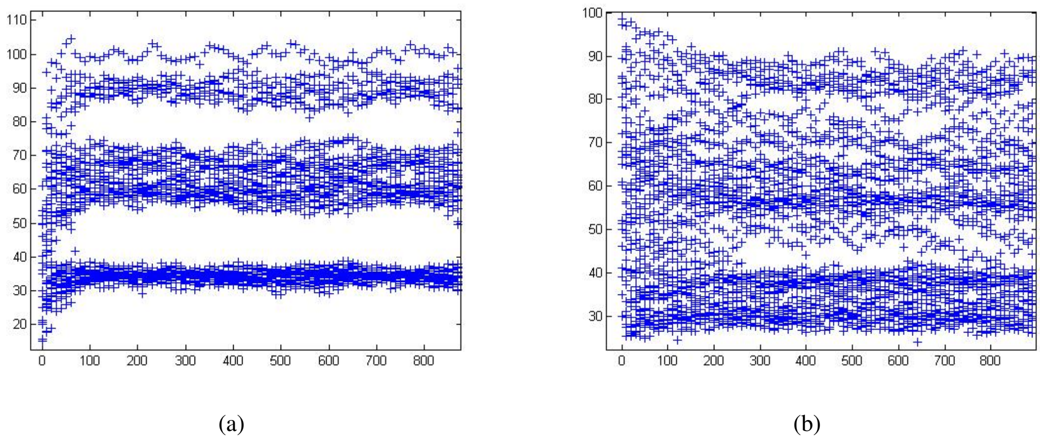


Figure 5.13: *Interrobot distances as a function of time, using potential controller (a) for setup 1 (b) for setup 2*

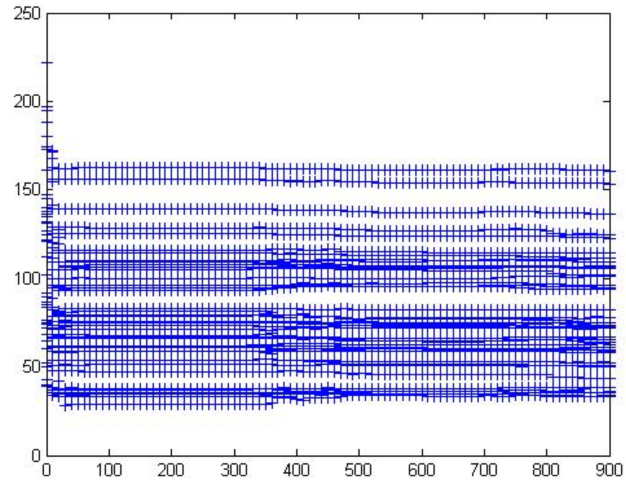


Figure 5.14: *Interrobot distance as a function of time. Setup 3 with potential controller and phase alignment controller*

5.7 Discussion

It is clear that the potential function controller is capable of obtaining hexagonal lattice formations with settling time ranging from 4 minutes to more than 15 minutes. Especially setup 1 and 3 show hexagons, whereas setup 2 shows more of nonsymmetrical equidistant formation. It was not a given that hexagonal lattice formation was obtainable since the robots sensors are spaced at 45°angle and there are blindspots in the sensing field. The angles in hexagonal lattice formations are 60°. Figure 5.11 gives a clue as to what happens. The robots without phase alignment control spin in the same place thereby compensating for the blind spots in the sensor field. The phase controller will counteract that spinning unless of course all robots are spinning in synchronized phase. To obtain a better synchronization a statemachine could be implemented to separate the potential function controller and the phase alignment controllers so they don not run in parallel. It is of course possible to just add better sensing, but that goes against the idea behind a cheap and simple platform. If a square lattices formation algorithm like the one suggested in [SSHH02]was implemented the robots might not be spinning at all since the angles of that formation is 90°. That would be a very interesting subject for further research on this platform.

5.7.1 Further Research

Because of the good simulation results the algorithm should be implemented on the real robots. Since the robot and communication models are quite accurate the transition should be fairly easy. The simulator can then be made more accurate by tuning the models further. The robots are equipped with radio communication making it possible to even compare simulation and real experiment in real time. Obstacles should then be introduced into the simulation to achieve a proper model of the environment.

Chapter 6

Implementation on Real World Robots

6.1 Introduction

Robotic science is often based on a set of sensors that gives information to the computer which calculates and decides a desired action from the information given. From that desire, it commands some actuators to execute the action whereby an input/output relation is created. As stated in Section **??** robotic science is most often developed and tested in a simulation environment. Working with real world robots there are many things to consider, when this relation between input and output is designed. The inputs given by the sensors are easily disturbed by the surroundings such as smoke or ambient light, depending upon the sensors chosen. Concerning the output there are forces like gravity and friction of the surface as well as friction in the transmission parameters, which can be difficult to calculate. In a simulated environment as described in Chapter 5 it is easy to neglect these parameters, since the behavior of the robot is determined through a model that we create ourselves. A simulation will furthermore provide a fast way of change parameters as well as solve a lot of practical problems like resetting and programming the robots, change batteries or breakdowns. The main reason why most scientists use simulation instead of real world robots is the complicated and time consuming process of actually building the robots. This paper describes the design of 10 new robots developed for the project. Furthermore it presents an implementation of the relation between the repulsive and attractive on the new robots.

6.2 Methods

Based on the analysis of the robots provided for this project in Chapter 2, it was decided to upgrade them through a redesign into a new version. The main reason for that decision was a great desire for a wireless connection to fulfill the methodology described in Chapter 4. Furthermore it was concluded that the mechanical design could be improved to avoid the robots clinging to each other. Prior to achieving finished robots, lies a lot of hardware testing to make sure that the right components are chosen, and a test bench of a prototype is often used. Our test bench, as illustrated in Figure 6.1 consist of two Crumb128 modules from Chip45 [chi], which uses the Atmel ATmega128 microprocessor providing the same functions as the robots.

The design of the new robots will be described next, from the perspective of the wanted functions. The design is based on the lessons learned from; the analysis of the provided robots described in Chapter 2, the demands of the desired methodology presented in Chapter 4 and the test of new hardware performed on the test bench.

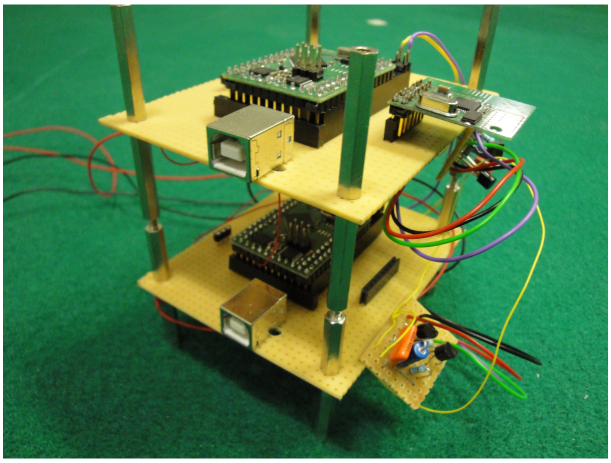


Figure 6.1: Picture of the testbench used in the analysis of the wanted functions of the new robots

6.2.1 Description of New Robots

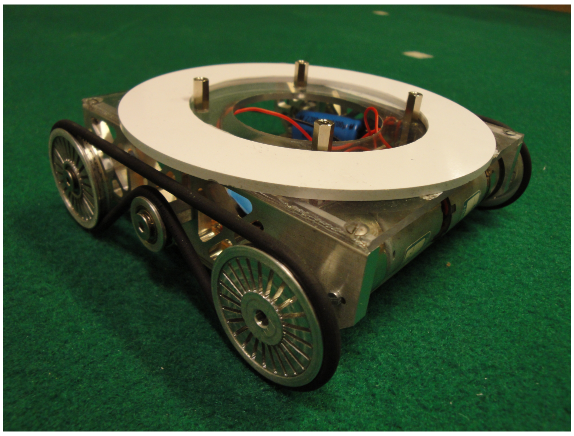
This project is a part of the groups working in the More Than One Robotics Laboratory at Aalborg University [Ngo]. The idea behind making these new robots is that they should be used in the future, by other students as well. Hence we need to consider demands on functions for future projects, that we might not make use of in this project. For that reason the robots must be well designed with both the electronic hardware as well as the mechanical design in mind. A description of the design based on the desired functions will now follow and Figure 6.2 illustrates the new robot design with and without the body.

(a) (b)

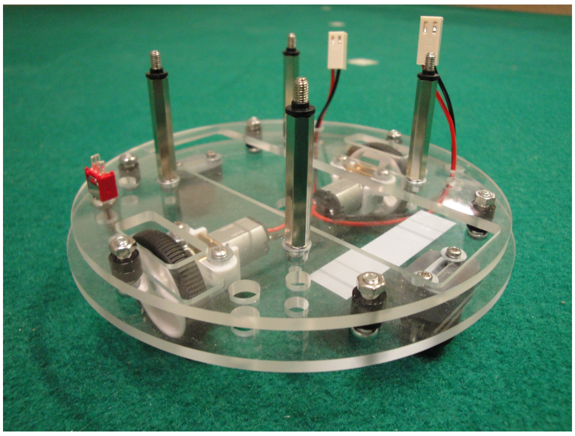
Figure 6.2: The new robots with and without the body

Mechanics

The new mechanical design is based on simplicity and adaptability. The frame is made of two layers of 3mm plexiglas in a circular shape, and a white tube is placed upon these to create the body of the robot. This body will also decrease the amount of noise of the IR receivers as well as reflect the signal used for measuring signal strength. On the old robots, the on/off switch was placed on the side of the frame, risking that a neighbour could hit the switch and turn it off by accident. On the new frame the switch is placed underneath avoiding this problem. Furthermore it is reachable with the body still on. Figure 6.3 illustrates the difference between the chassis of the old robots compared to the new robots.



(a)



(b)

Figure 6.3: *(a) Chassis of old robots. (b) Chassis of new robots.*

Propulsion

On the old robots the use of a heavy aluminum chassis made large motors necessary for them to be able to run fast enough. Because of the new lighter design a set of two smaller motors has been chosen from Pololu Robotics and Electronics [Pol] with gear ratio of 30:1. As a result the motors are drawing less current from the motorcontroller board providing longer runtime on our batteries. The batteries used are the same as in the old robots, namely 8 cells of AA rechargeable NiMH batteries ranging from 2500 to 2700 mAh resulting in around 9,6v when fully charged. The board is also designed to work with 2 cells of Lipo giving 7.4v, which will be an alternative to the NiMH cells in the future. Lipo batteries are much lighter and smaller relative to how many mAh they provide; making it possible to get either lighter robots or longer runtime.

Boards

The frame provides four holders for boards able to stack them on top of each other; hence boards can be added or taken out very easily, as seen on 6.2(b). Furthermore it gives the possibility to develop new boards in the future, with new functions to the robotic platform without any changes in the mechanical design. Each robot consists of three boards and the bottomboard controls the motors as well as a 2.4ghz radio module and a Bluetooth module. The middleboard and topboard are almost identical new versions of the old sensorboards, and are used for measuring the distance, and control IR communication between the robots.

Each board is controlled by the use of an ATMega128 microcontroller from ATMEL [Atm]. The microprocessor was chosen based on the fact that it was used in the old robots, which have been proven to fulfill the demands of the operations needed. Hence it is known technology around the laboratory, making it very easy to correct errors and reuse and test formerly developed code.

Interconnection

The communication between the boards is done with an interconnection using Two Wire Interface (TWI), which is also known as i^2c. The standard uses only two wires of bidirectional open-drain lines to communicate. The two lines are a Serial Data Line (SDA) and Serial Clock (SCL), pulled up with resistors as illustrated at Figure 6.4. With a 7bit address space where 16 addresses are reserved it results in 112 nodes, designating every node its own ID number. The bus has two roles for nodes in the network, The master is the node that creates the clock and addresses the slaves, while the slaves receive the clock line as well as the address. The bus is able to use more than one master, but in our case we keep it simple and use only one master. The bottomboard is designated as master and the two sensorboards as slaves. Data can only be transmitted from the master to any of the slaves or from any of the slaves to the master. Hence the two slaves, in this case the two sensorboards, cannot communicate with each other directly. The interconnection also provides power to the middle and topboard, from the bottomboard.

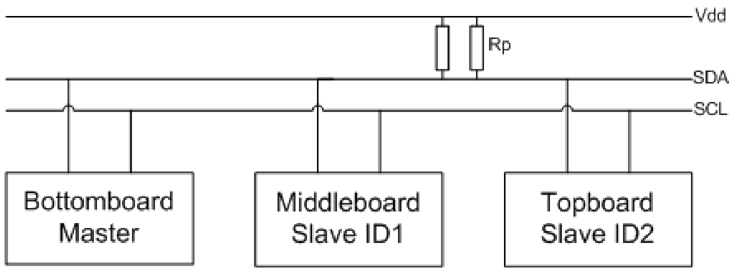


Figure 6.4: *Diagram of the implemented TWI bus*

USB serial connection

Each board provides an USB serial connection able to connect with a PC to illustrate internal data of the microprocesser, through either Matlab or a terminal. This function is very useful when testing and debugging code, to make sure that the right data is send and receive throughout the different connections. The ATMega128 provides two UARTs where UART1 is dedicated for use with the USB serial connection on each board. The chosen baudrate is 38.000kb/sec. On the bottomboard is the UART1 also used for the radio module, which is switched between by moving a jumper.

Measurement of light strength

To measure the distance to objects the middleboard is chosen. Like the topboard it is mounted with eight sets of sensors each consisting of two transmitters and one receiver. The receiver is an IR photodiode which is variable in the current and not the voltage as the transmitters. This means that a converter from current to voltage is needed before it can be used as an input to the microcontroller. When measuring a strength, we need the value of the input from the receivers to be variable because the distances to other robots and obstacles are variable. Hence we use the ADC port on the microcontroller which uses a signal in the range from 0 to 5 volts to map into a distance.

IR Communication

The robots should be able to communicate to other robots using the infrared sensors. It sends out a robot ID and sensor number. The topboard is dedicated to use for communication and the new version of the sensor boards are able to broadcast on all sensors at the same time, by setting a pin high on the microcontroller. The IR sensors are using UART0 providing a serial interface for communication.

Rotating mirror

The old robots provided an IR sensor able to scan the environment using a rotating mirror on top of the robots. It was decided to leave this function on the sensorboards, even though this project is not going to use it, former projects have used this and future projects could gain great benefit from this function. It uses the Rx channel while the serial USB connection uses the Tx channel of UART1.

Ambient Light

Just as the rotating mirror is this a function which is not used in this project, but is left on for future purposes.

Wireless

The need for a wireless connection to each robot is important for this project in several ways. As explained in Chapter 4, programming robots is a very time consuming task when testing algorithms. When a swarm of 10 robots, each having three microprocessors that need to be programmed, the task becomes too large and a new method must be taken into use. By use of a wireless connection it is possible to program all the robots at once from a basestation computer. For this operation a Bluetooth module AMB2300 [AMB] is chosen because of the following: Fast transmission speed is needed because the compiled programs can be very large. The Bluetooth standard is very stable compared to other radios, which is needed to avoid errors when writing to the microprocessors. Bluetooth has a relative short range of connection, but it is very likely that the robots will be placed near the basestation when reprogramming which makes Bluetooth ideal for the job. The module uses a serial connection of which UART0 of the bottomboard is chosen.

Instead of reprogramming all the robots it is even possible to change variables on the fly. This is very useful when tuning an algorithm by changing parameters such as distance between robots, goal location and the phase alignment. Furthermore to send a new command to the formation such as: Go, Stop, change formation etc. It also gives the feature of receiving values from for example sensors directly to the basestation screen, which can be very useful for analysing if a given sensor is broken. For these operations the Bluetooth module is not suitable because of its low range of connection and its heavy power consumption. Hence a iDwaRF-168 2.4ghz radio module from Chip45 [chi] is chosen, which is connected to UART1 on the bottomboard

6.2.2 Implementation

The implementation of the formation algorithms is based on the simulation implementation described in Section 5.4. Hence the following will describe where on the robot the functions are placed and leave the explanation of the algorithm behind the relation between attractive and repulsive force to the description of the simulation. To get an overview of the robotic design Figure 6.5 will illustrate the use of functionalities and the communication between the different modules.

The red lines outline the three boards. The green boxes show the functions provided from the hardware in form of either an input or an output. The blue boxes are modules of code that take care of a functionality and handles inputs from sensors and output to actuators. The blue lines between modules are the handling

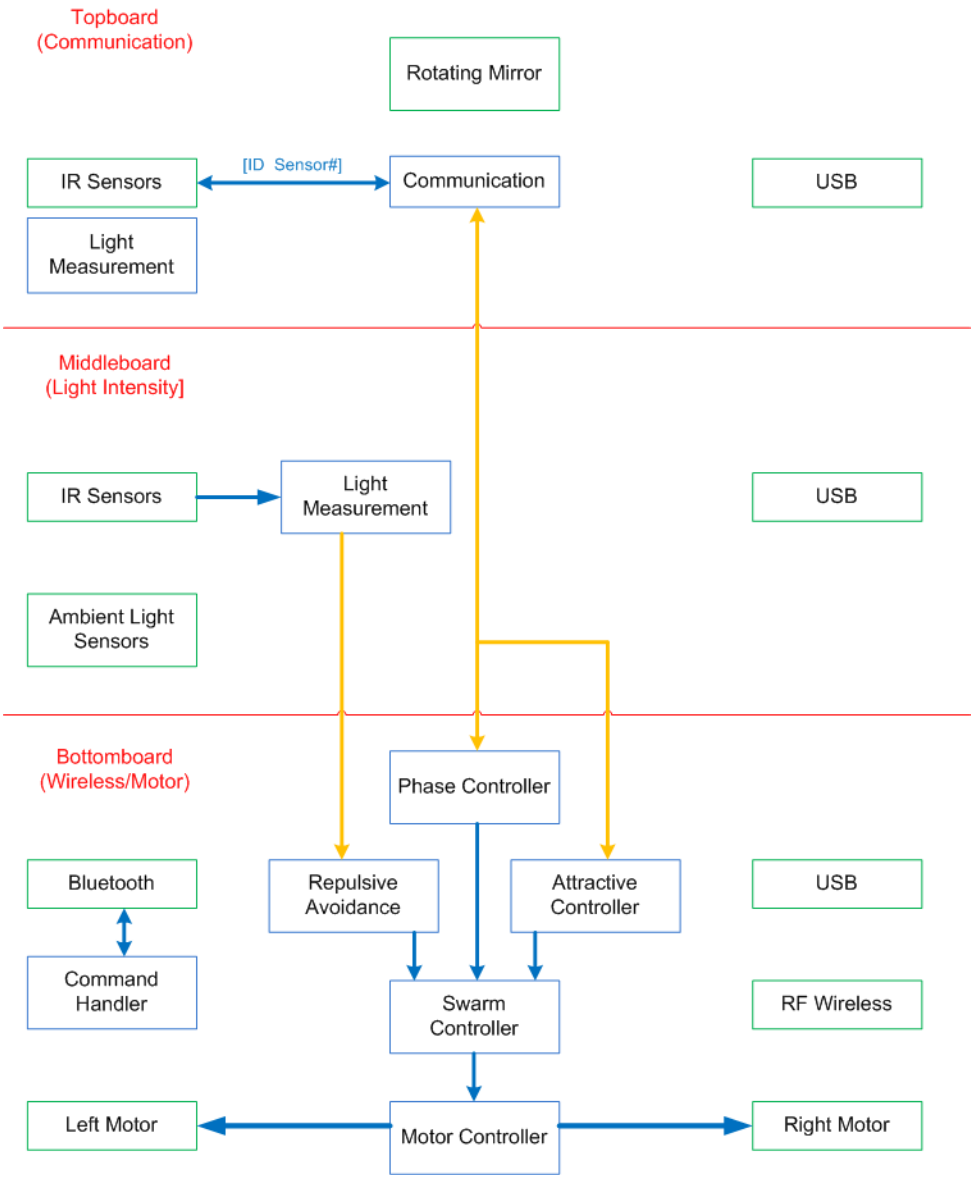


Figure 6.5: *Diagram*

of variables, providing information between the different modules within the same microprocessor. The orange lines are the information send from board to board through the TWI. Note that the rotating mirror, ambient light sensors and the RF Wireless module are objects not used, but illustrates the functions to use in future projects.

We want to use the topboard for communication between the robots and as illustrated on the topboard in Figure 6.5 a Communication module is used to handle both transmitting and receiving. The robot transmits an ID of the robot and the number of the sensor used in the transmission. If a robot is near it receives the same information from the neighbour which is sent to the phasecontroller running on the bottomboard.

The ID and sensor number is also sent to the attractive controller for use to identify whether it is an object or another robot detected. If the ID is different from its own ID it is a fellow swarm robot and an attractive force is created towards the neighbour. We then need a repelling force that will neutralize the sum of both forces when the robot reaches the wanted distance.

The middleboard is dedicated for measuring the distance to any object around the robot. The value of the distance is then sent to the repulsive controller module on the bottomboard. If the distance is less than the wanted value it will create a repulsive force larger than the attractive force resulting in the robot moving away from the neighbour. If the distance is larger than wanted the repelling force will be less than the attractive force resulting in the robot will move towards the neighbour. The obstacle avoidance is totally independent on whether the object is a robot or a wall. If the robot is too close to an undefined object it creates a repulsive force meaning that there is no attractive force created since there is no ID input to the topboard. This is a very simple implementation of the obstacle avoidance, hence that the wanted distance between the robots is also the same minimum distance to any other object. If a these values are wanted different, the obstacle avoidance should be implemented as a part of the attractive controller.

The repelling and attractive forces are handled in the Swarm Controller module which convert the vectorforce to an actual force on the two motors resulting in propulsion in either direction towards or away from a neighbour depending on the distance between them at the given time.

As seen on Figure 6.5 the Bluetooth module is connected to a Command Handler. This handler takes care of the input from the Basestation through Bluetooth. To compare testdata if multiple tests are done, it is crucial that the initial conditions are similar. One command that is very important in this case is the ability to start all robots simultaneously, which can be done through the Bluetooth connection.

6.2.3 Test Description

Caused by a late arrival of the new electronic boards, the time for tuning the parameters in the implementation of the algorithm was limited, hence a proof of concept implementation of the relation between the attractive and repulsive forces was created. The implementation was based on the method described above and the lessons learned from the simulations in Chapter 5.

Based on the knowledge from some preliminary test of the infrared communication between the robots, it was concluded that when a robot should both transmit and receive the statistic chances of successful communication is reduced. Hence a better implementation of the transmit and receive relation should be implemented in the future to improve the chances of communication. To compensate for this simple implementation it is desired to reduce the risk of errors. For that reason a simple test scenario was created able to prove the concept of attract and repulse, meaning that the phasecontroller is not implemented and tested on real world robots at this state.

For a robot to be attracted to another robot it only needs to receive the ID of the neighbour. A stationary robot was programmed to broadcast its own ID on all eight sensors. The moving robot is then expected to be attracted to it, until the repulsive force and attractive force neutralize each other. Hence the robot has reached the wanted distance to the stationary robot. Two different tests was designed, one to test the attractive force alone and one where the attractive and repulsive relation is tested.

First we would like to test the use of only the attractive force and by setting the repulsive parameter to 0 the algorithm will always return a repulsive force of a zero value. Hence the resulting force is the attractive force created from the communication between the moving robot and the stationary robot. The moving robot is placed randomly around the stationary robot with the distance of approximately 100 cm. It is expected that the robot will move in a fairly straight line towards the stationary robot eventually hitting it because of the missing repulsive force.

Next the relation between the attractive and repulsive force is tested. The parameter of the repulsive force is set to -1 meaning that the force created from the measurements of the distance to the stationary robot will be subtracted from the attractive force. The moving robot is placed randomly around the stationary robot within 100 cm, and it is expected that it will establish connection and stay close to the stationary robot without running away. The test runs for five minutes. To evaluate whether or not the expectations of the tests are meet, we use Swistrack to collect the coordinates of the robots as described in Section 2.1. The results in the following are representations of the trajectory of the moving robot.

6.3 Results

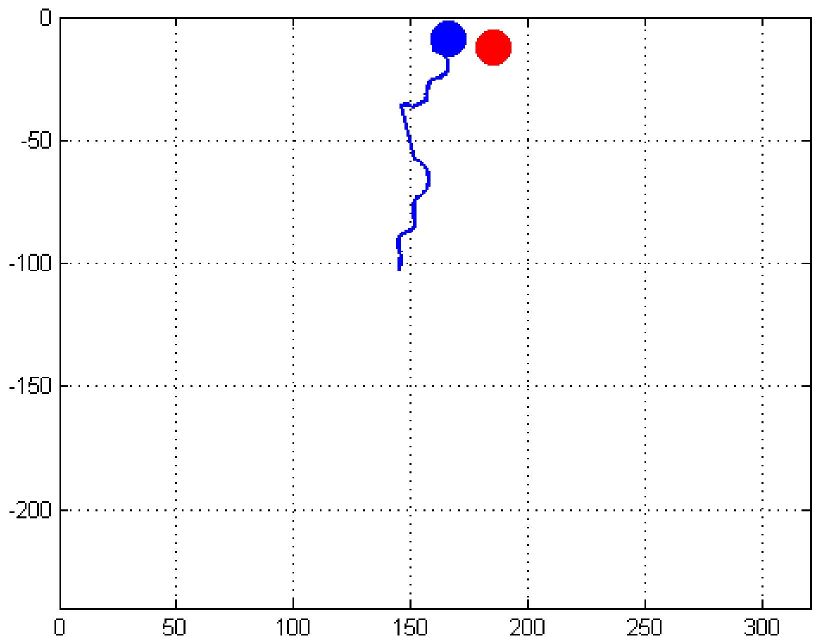
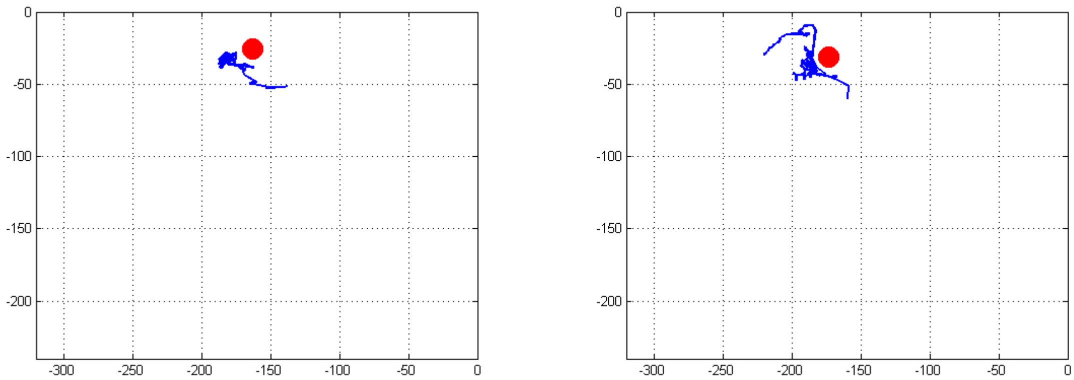


Figure 6.6: *Plot presents the trajectory of the moving robot illustrated by the blue dot. The red dot illustrates the stationary robot.*



(a) (b)

Figure 6.7: Attractive and repulsive forces implemented. The blue line illustrates the trajectory of the moving robot flocking around the stationary robot illustrated by the red dot.

6.4 Analysis

In the section above are one result from the test of the attractive force and two results from testing the relation between the two forces.

6.4.1 Test of the Attractive Force

As seen on the graph of the robots trajectory the robot clearly moves towards the stationary robot. The test was repeated 10 times where the initial position and phase of the moving robot was random within 100 cm to the stationary robot. The behavior of the moving robot was the same in all 10 attempts, resulting in it hitting the stationary robot as expected.

6.4.2 Test of Attractive and Repulsive Relation

The results illustrate the trajectory of the robot flocking around the stationary robot. Even though the algorithm contains a repulsive force the robot hit the stationary robot a couple of time during the test, but was generally good at avoiding it. Just as in the attractive test it was repeated 10 times resulting in the same behavior.

Both tests resulted in the expected behavior. From these results we have proven that the concept is working and by improving the implementation it should be able to achieve formations as it is illustrated in the simulations in section 5.5. In the following it is discussed how to achieve this.

6.5 Discussion

From the analysis of the results it was stated that we are able to create a repulsive and attractive force using measuring of distance and communication between robots. In the following we will discuss how to improve the simple implementation to achieve formation of multiple robots.

The test of the implementation of the algorithm of attractive and repulsive relation on real world robots is based on only one robot moving towards a stationary broadcasting robot. In a formation of multiple robots, redundancy of the robots is desirable, meaning that all robots should have equal functionality. Hence all robots should be able to both send and receive data as described in Subsection 6.2.2. To increase the chance of the robot receiving at the same time as the neighbour is sending in the direction of the robot, a new implementation of the send/receive cycle is needed. The implementation in the test already used broadcasting on all sensors which reduces the error significantly. But when the phasecontroller is wanted implemented it needs to know the number of the sensor which is sending. Hence it is not possible to just broadcasting the robot ID on all sensors.

If both robots are started at the same time and tranceiving on one sensor at the time, using sensors 1 trough 8, they will never be able to communicate when facing in the same direction. Not unless they are facing opposite directions or started at different times, is there a chance of communication present. This is true since the neighbour is transmitting in the opposite direction of the robot as it is receiving towards the neighbour. By sending the data on a random transmitter it increases the chances of a receiver and transmitter reaching each other resulting in a successful transmission.

The middleboard is used for measuring the intensity of send IR data reflected on an object and create a repulsive force from that information. In the test implementation, the board just sends out useless information to measure on. Because the topboard is not allowed to only broadcast the robot ID, the middleboard could take over this function and broadcast ID of the robot. This will increase the chances of creating an attractive force.

phasealignment!

parameters fitting.

robot hitting stationary

not straight line (attractive)

not steady state (relation)

Part III
Tests and Conclusion

Chapter 7

Discussion and Conclusion

This project has had a very broad perspective.

- Analysing and implementing robotic swarm and formation algorithms in both simulation and on real robots.
- Analysing and developing a methodology for developing robotic swarm and formation algorithms.
- Developing and producing new robot hardware and simulink models of this hardware.

7.0.1 Methodology

A unified methodology for developing robotic swarm and formation algorithms was developed and presented at the RISE 10 workshop in Sheffield UK. The methodolgy has formed the base for design and development of hardware and simulator. Integration of the hardware, simulator and a GUI is not completed.

7.0.2 Formations and Algorithms

Formation algorithms where analysed an simulated. A reading group was formed to gather and share knowlegde by reading and presenting papers on various fields of robot swarms. This laid the foundation to test other peoples results on the platform developed in this project. The simulation produced some ver

7.0.3 Hardware and Software

10 new Robots have been designed and produced. The design is a further development on the existing robots. A lot of time has been spend on the analysis and test of radio modules and design of PCB, since the design has to be durable. These robots are not first prototypes and should last much longer than this project.

The huge amount of hardware work has pushed deadlines to a point where the total envisioned system has not been completed. Large key components have been finished. That is a simulator in Simulink, new robots supporting wireless communication and improved IR sensing and a tracking system making it possible to track the position of each robot. What is left is the integration of these components and the addition of a GUI.

This project group succeeded with the following:

- Developing a unified methodolgy for Swarm Algorithm development and evolution, and presenting it on a RISE workshop.
- Developing a simulator in Matlab Simulink.
- Developing a new robot platform. Designing PCB
- Proving via simulation that hexagonal lattice formations can be obtained by the new robots.
- Proving via real life experiments that virtual potential field can be implemented on the new robots.

In 3 a set of requirements to formation was set. Since formation only has been carried out in simulations the conclusion is based on the results in chapter 5 §1 The deviation of the distance between the neighbours should be within 10cm.

This requirement can be held acording to figures 5.12 and 5.13(a). However on figure 5.13(b) the requirement is clearly not held when taking the last requirement into account.

§2 The robot should be able to align its phase to the other robots within 45°.

This requirement is clearly not held. Figure 5.11 show no allignment of phase

§3 The robots should meet the requirements for phase and distance within 15 minutes.

Since the figures used to argument are plots of 15 minutes length the conclusions at the to previous requirements are sufficient.

The developed hardware and software is going to be used in future project on Aalborg University. Some sugested improvements could be:

- Ad phase tracking to the tracking system. Since the phase is a part of the simulator it would be preferable to be able to compare measurments and simulation data
- Switch batteries on robots to LiPo. The current NiMh batteries suffers from a high inner resistance when large current are drawn. This results in the brown out detector reseting the microprocessor.
- A better software implementation of transmit algorithm. At the moment the transmit algorithm is synchronized meaning that robots may never see each other. Adding a clever form of random would remove synchrone blind spots.

Bibliography

[AMB] AMBER. Internet, June. http://www.amber-wireless.de/.

[APP02] Tamio Arai, Enrico Pagello, and Lynne E. Parker. Editorial: Advances in Multi-Robot Systems. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, 655-661*, 18(5), 2002.

[Atm] Atmel. Internet, June. http://atmel.com.

[BAF$^+$] Laura Barnes, Wendy Alvis, MaryAnne Fields, Kimon Valavanis, and Wilfrido Moreno. Swarm Formation Control with Potential Fields Formed by Bivariate Normal Functions.

[BC] T D Barfoot and C M Clark. Motion Planning for Formations of Mobile Robots.

[BHKP] Spring Berman, Adam Halasz, Vijay Kumar, and Stephen Pratt. Algorithms for the Analysis and Synthesis of a Bio-Inspired Swarm Robotic System.

[CFK97] Y. UNY CAO, ALEX S. FUKUNAGA, and ANDREW B. KAHNG. Cooperative Mobile Robotics: Antecedents and Directions. *Kluwer Academic Publishers, Boston. Manufactured in The Netherlands*, 1997.

[chi] chip45. Internet, June. http://chip45.com.

[INS01] Luca Iocchi, Daniele Nardi, and Massimiliano Salerno. Reactivity and Deliberation: A Survey on Multi-Robot Systems. *MAS, LNAI 2103, pp. 9-32*, 2001.

[LSJK09] Casper Hillerup Lyhne, Erik Baasch Sørensen, Lars Jespersen, and Marc Westring Krogen. Network protocol stack for robots in a scalable swarm. *AAU SEMCON 09*, 2009.

[MD] James McLurkin and Erik D. Demaine. A Distributed Boundary Detection Algorithm for Multi-Robot Systems.

[Ngo] Trung Dung Ngo. Internet, May. http://vnbotics.blogspot.com/.

[Par08] Lynne E. Parker. Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems. *JOURNAL OF PHYSICAL AGENTS*, 2(1), MARCH 2008.

[Pol] Pololu. Internet, June. http://www.pololu.com/.

[Sah05] Erol Sahin. Swarm Robotics: From Sources of Inspiration to Domains of Application. *Swarm Robotics WS 2004, LNCS 3342, pp 1020*, 2005.

[Sha] Amanda J.C. Sharkey. Swarm Robotics and minimalism.

[Sie05] Gabriel Hugh Elkaim Michael Siegel. A LIGHTWEIGHT CONTROL METHODOLOGY FOR FORMATION CONTROL OF VEHICLE SWARMS. *IFAC*, 2005.

[SSHH02] William M. Spears, Diana F. Spears, Jerry C. Hamann, and Rodney Heil. Distributed, Physics-Based Control of Swarms of Vehicles. 2002.

[TCGS08] Ali E. Turgut, Hande Celikkanat, Fatih Gökce, and Erol Sahin. Self-Organized Flocking with a Mobile Robot Swarm. *AAMAS*, 2008.

[Win] Alan FT Winfield. Swarm Robotics and minimalism.

Part IV
Appendix

Appendix A

Cd Content

The appending CD contains the report in both PDF and Latex format, all used software to run the robot, papers used in the studygroup, The Simulink model of the intire system used for simulation, Orcad drawing of the developed robots and Videos of the real world robots running. The primary folders and secondary folders on the CD is shown here.

- Report
  - Report PDF
  - Report Latex
- Robot Software
  - Topboard
  - Middleboard
  - bottomboard
  - Stationary Broadcasting
- Papers from study group
- Simulink model
- Orcad
- Videos