

# Swarming

Research document

Version 1.0

March 11, 2016

M. van Wilgenburg, W. Mukhtar, E. van Splunter, M. Siekerman, T. Zaal and M.

Visser

## List of Figures

1	Worst case scenario for two Zebro light robots at close range . .	5
2	Maximum amount of robots within the close range radius, the maximum amount results in 130. . . . .	7
3	Data conditions . . . . .	8
4	Structure of a TWI message . . . . .	8
5	Unit circle where points P and Q are mirrored on the X-axes . .	11
6	Relative location interpretation . . . . .	12
7	N1 Mirrored interpretation . . . . .	13
8	Triangulation using three stationary beacons . . . . .	14
9	Determining the angle with three stationary beacons . . . . .	15
10	relative localization system using relative distance and orientation	16
11	Distance versus RSSI plot showing inverse non linear relationship.[3]	17
12	Distance versus RSSI plot showing lack of reliability when tested in a different direction.[3] . . . . .	18
13	Distance versus RSSI plot showing lack of reliability when tested in a different direction.[3] . . . . .	18
14	In sequential equivalent time sampling a singel sample is taken for each recognized trigger after a delay which is incremented after each cycle.[6] . . . . .	20
15	Example equivalent time sampled signal.[6] . . . . .	20
16	Accuracy RSSI vs TOF.[4] . . . . .	21
17	An example of a wireless mesh network with 5 clients and an op- tional internet connection. The connections made in this example are random. . . . .	25
18	Robot met mechanische proximity sensor . . . . .	27
19	The left graph shows the average PSD from 50 echos generated by a plastic botthle from a distance of 100cm, the average envelope is generated using 50 echos of the same bottle . . . . .	28

## List of Tables

1	Localisation specification . . . . .	6
2	Specifications number of units . . . . .	7

## Contents

<b>1</b>	<b>What do we consider swarming?</b>	<b>4</b>
1.1	Situational sketch . . . . .	4
1.2	Precision of localization . . . . .	4
1.3	Number of units of the Swarm . . . . .	6
<b>2</b>	<b>Protocol</b>	<b>7</b>
2.1	Hardware . . . . .	7
2.1.1	TWI . . . . .	8
2.1.2	SPI . . . . .	9
2.1.3	UART . . . . .	9
2.2	Software . . . . .	9
<b>3</b>	<b>Localization</b>	<b>10</b>
3.1	Angle determination . . . . .	10
3.1.1	Goniometric angle determination . . . . .	10
3.2	Relative Distance . . . . .	17
3.2.1	Received Signal Strength . . . . .	17
3.2.2	Time Of Flight . . . . .	19
3.3	Conclusion . . . . .	21
<b>4</b>	<b>Propulsion,Actuators and Effectors</b>	<b>22</b>
4.0.1	Effectors . . . . .	22
4.0.2	Actuatoren . . . . .	23
4.0.3	Aandrijving . . . . .	23
<b>5</b>	<b>Swarm Communication</b>	<b>24</b>
5.1	Proximity sensor . . . . .	26
5.1.1	Ultrasonic sensor . . . . .	26
5.1.2	Lidar sensor . . . . .	26
5.1.3	Mechanische sensor . . . . .	27
5.1.4	Sensor choice for the robot . . . . .	27
5.1.5	Obstacle discrimination with an ultrasonic sensor . . . . .	28
<b>6</b>	<b>Bibliografie</b>	<b>29</b>

# 1 What do we consider swarming?

Swarming is a wide concept and can be achieved in many forms. In this section will be defined what we consider swarming during this project. Many choices made here will be up for discussion, but these choices have to be made so the project can be successfully specified. For example, no clear minimal number of units is specified for a group of robots to be called a swarm. When specification have to be exact and testable this becomes a problem. The following subjects will be discussed: number of robots and scalability, precision of localisation. Its important to keep in mind that the choices made in this section, mostly consist of educated guesses. These guesses will later be confirmed by testing.

## 1.1 Situational sketch

To be able to make any assumptions we will first sketch a situation. We propose a test situation where a swarm of eight robots need to explore a room as efficient as possible. They will need to find a certain object and when the object is found by one of the units, the others will have to come to the objects location. For the robots to search efficiently they will need some type of "close range" relative localisation, so they can spread evenly across the room without bumping into each other. Speed is also an important factor when considering the frequency of distance measurements. For this sketch we chose an average speed of 3,6km/h or 1m/s, this a bit slower than average walking speed and seems realistic for smaller robots. The robots are programmed to keep a distance of 5cm from each others outer boundaries. And should keep a absolute minimal distance of 2cm.

## 1.2 Precision of localization

An obvious answer for the question "How precise does the localisation need to be?" would be "As precise as possible!". However with swarming robots it might not need to be precise at all. This depends mostly on what the swarm needs to accomplish. For the situation we proposed the localisation needs to be precise at closer range but can be less accurate when further away. We choose to define "close range" as 0 to 3 meters, and "long range" as 3 to 25 meters (further if possible). The big difference between the two is the margin of error that is allowed.

Close range localisation has a smaller margin of error because the robots will need to be able to move precisely, to perform certain tasks without bumping into each other. To determine the precision needed, size of the robots matter. The project group of the Zebro light set out to make a robot with maximum dimensions of 300x 200 x 60 mm [?]. When the swarming module is placed in the middle of this platform the the absolute minimum distance between the two modules will be 20cm from middle point till middle point. Considering that some deviations might occur the minimal measuring distance or "dead zone" is chosen to be 15cm. The worst case scenario is shown in figure 1 the maximum deviation is calculated with by dividing the absolute minimal distance (2cm) by the distance between the swarming modules (22cm). This results in a maximum deviation of 9% in the close range distance measurement. The relative angle to another swarming module is needed to successfully implement localisation. At

close range the localisation will mostly be used so the robots don't collide, or to implement swarming behaviour like trail following as discussed in the problem definition. Limited information about the relative angle would be enough to perform these tasks. We propose angle measurement at close range to have a resolution of  $45^\circ$ . In field testing will have to validate these assumptions. The speed of the robots determine the frequency at which the close range localisation should take place. The robots in this sketch move at a maximum speed of 1 m/s, the worst case scenario is that the robots will approach each other head-on. This makes their combined speed 2 m/s. The robots have been programmed to keep a minimal distance of 5cm from each other. Dividing the minimal distance by the combined speed will result in 40Hz which is the minimal required sampling speed.

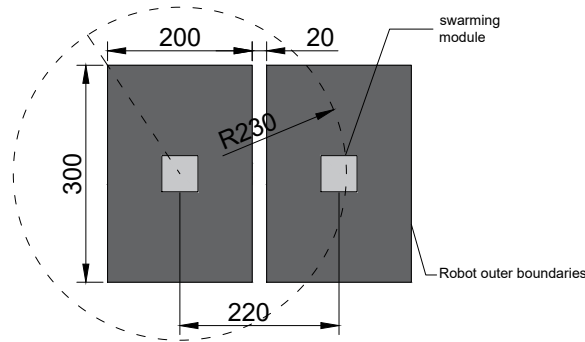


Figure 1: Worst case scenario for two Zebro light robots at close range

Long range localisation will only be used to keep members of certain groups in the swarm together. When the units are working together on a task, they will be using the close range localisation. A situation might occur where a members of a group of members of the swarm will have to complete a task outside of the range of the close range localisation. When they are done and they want to return to the swarm the long range localisation will be used. As discussed earlier the long range localisation will start at three meters. But it has to get the unit in close range. Therefore the minimal measuring distance will be chosen at two meters. The distance measurement will need to be precise enough to detect when it reaches the overlap between 2 and 3 meters. This maximum deviation for this to be true is 1 meter. Long range localisation will actually be a long range distance measurement. The relative angle between modules will not be needed. When an unit needs to return to the rest of the swarm a algorithm can be developed that finds it way back by looking at the changing distance. When the robot is trying to determine the right direction, the update frequency is important. At far range we would assume that a robot has to move one meter to determine if its getting closer or not. With the robot moving at 1 m/s the minimal update frequency is 1Hz.

Summed up the specifications of localisation are:

	<b>Close range (0 - 3 m)</b>	<b>Long range (3 - 25m)</b>
Deadzone distance	0,015 meters	2 meters
Deviation (distance)	+/- 9%	+/- 1 meter
Angle Resolution	45%	-
Update frequency	40Hz	1Hz

Table 1: Localisation specification

### 1.3 Number of units of the Swarm

As discussed before in the problem definition, its hard to put an exact number on what number of units it required to form a swarm. We mentioned "a high number of units" and that a minimum number of units is hard to justify. We will look at what the preferable amount of units would be to demonstrate the swarming module. Also we will look at a minimum amount and a maximum. Important to remember is that the swarm must always be scalable, so no static number of units will be defined.

Robots in a swarm have limited capabilities on their own, thus they need to work together to perform a complicated task. When one unit breaks down the others should still be able to complete the task. With this in mind the absolute minimum units required for a swarm is chosen to be 3. When one breaks down the other two can still complete the task faster/better than one could do on its own. It would be preferable to make the total swarm endlessly scalable. When developing the swarming module no maximum number should be defined to the swarm. Hardware limitations will later determine a maximum. However a maximum of units can be calculated, that one unit has to be able to detect at close range. The detection surface can be viewed as a circle around of unit with a radius of 3 meters. The radius for the circle is also calculated for the area where robots will avoid each other. This is done by taking the furthest point seen from the center of the robot and then adding 0,05m (the range where robots will avoid each other), this circle is shown in figure 1. The maximum amount of robots in this range can then be calculated using a simple tool, the result is shown in figure 2. The end result is a maximum of 130 robots.

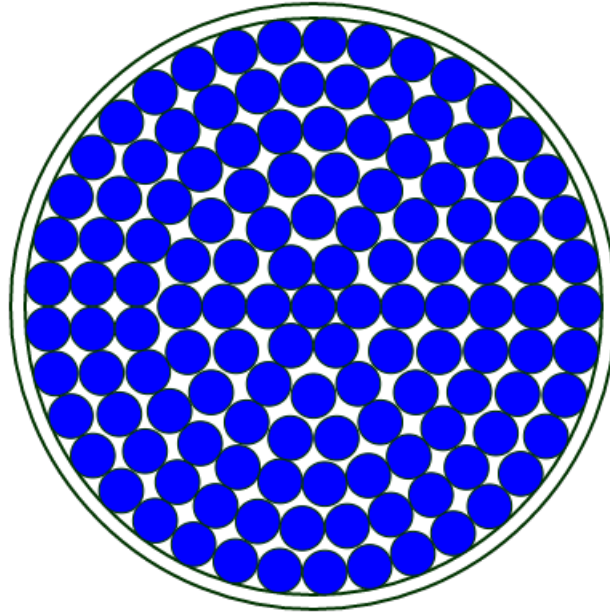


Figure 2: Maximum amount of robots within the close range radius, the maximum amount results in 130.

For the sake of demonstration we would like to recreate the "feel" of a swarm. This means a higher number of units is required. We chose this number to be between 6 and 12 units depending on the time and budget.

	minimum	maximum	maximum (close range)	Preferred (demonstration)
Number of Units	3	$\infty$	130	6 to 12

Table 2: Specifications number of units

## 2 Protocol

### 2.1 Hardware

For the internal communication between the different systems there are multiple protocols to choose from to transmit data. every subsystem with a microcontroller should be able to communicate with the rest of the system. Because every system should be able to share their information it is necessary to have a multimaster system which allows every system to share their information without being dependent on a single master. To see which protocols are viable multiple protocols will be compared.

Building a modular robot consisting of subsystems, there must be a method to let these "*subsystems*" communicate with each other to exchange relevant information.

Every subsystem features a micro-controller which must be capable of transmitting and receiving data.



### 2.1.1 TWI

The name TWI stands for "Two wire interface" which strongly resembles Phillips' protocol  $I^2C$ . Like the name suggests the bus is build up using two wires, one is used for the clock signal and the other is used for the data signal. Each of these wires is connected to the powerline through a pull-up resistor so every subsystem connected to the line can pull the line down which other systems can see. TWI can be used as a multimaster system which means every subsystem can send it's data without being dependend on a single master.

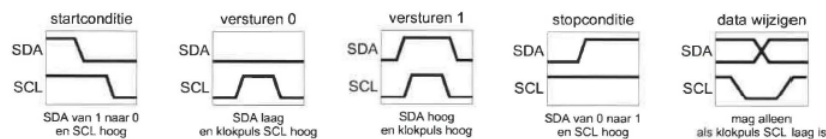


Figure 3: Data conditions

In the above figure multiple data and clock combinations are shown, these combinations define the message start, sending a zero, a one and a stopbit. The system that wants to send information starts a message bij changing the datasignal from one to zero on a high clocksignal. All systems that are listening now know one of the systems is going to send a message. Data can be send by changing the dataline from zero to one or the other way around during a low clocksignal, the other systems will read this bit when the clocksignal becomes high again. Once the system is done sending its message it will let the other systems know by sending a stopbit.



Figure 4: Structure of a TWI message

In the above picture the structure of a message is shown. As explained before the message starts with a startbit after which a small header follows. The header contains information about who is going to be adressd by the message and if you want to read or write. After this the adressd system will respond with an acknowldage to make sure the information has come across. When the acknowldage is recieved the system will start sending data, followed by and acknowldage from the reciever after every recieved byte of data. when the system is done sending all of its data and recieved the last acknowldage it will send the stopbit which means all communication has ended.

TWI has a few benefits, because it is a simple serial protocol it only uses two wires. this saves a lot of space in the hardware design. Also, it has the possibility to be used as a multimaster system. The speed of the communication depend on the frequency of the controller that will be used.

### **2.1.2 SPI**

SPI or Serial Peripheral Interface is, like its name suggests, a serial interface. This interface can be connected in two ways, each having its own advantage. How this works will be explained using the following figure.

### **2.1.3 UART**

## **2.2 Software**

### 3 Localization

Building a swarm of robots, location awareness may be required to ensure an more efficient operation. Implementing location awareness into a swarm of robots could result in better cooperation. For example: exploration based tasks can be executed on a much lower time scale. With location awareness the swarm can spread out evenly making sure that every part of the perimeter will be explored thoroughly.

With location awareness it is also possible to divide the swarm into multiple groups. Because the location of every individual is available these individual groups can be formed very quickly by selecting the nearest unit. This also enables quick assists when an individual robot gets stuck or needs to execute a task that require more than one unit.

For mapping the various positions of all robots in a swarm a localization technique needs to be implemented. The robots can use information as relative position and relative orientation of their neighbours and fuse the information collected by other robots to interpret the relative location. With relative position the units can define the distance relative to each other. When all robots in the swarm are aware of the distance opposite to each other, a map of the current formation can be created. However this can result in multiple solutions because there are no orientations involved.

#### 3.1 Angle determination

Determining relative orientation opposite to each other can be done in various ways. some methods involve larger limitations than others. In general angular measurements are done using the goniometric equations.

##### 3.1.1 Goniometric angle determination

The goniometric system shows how the different angles of a triangle within a circle can be calculated. Determining angles of a triangle within a circle using goniometric formulas mirroring can occur this happens because of the following relation between the angles see figure 5.

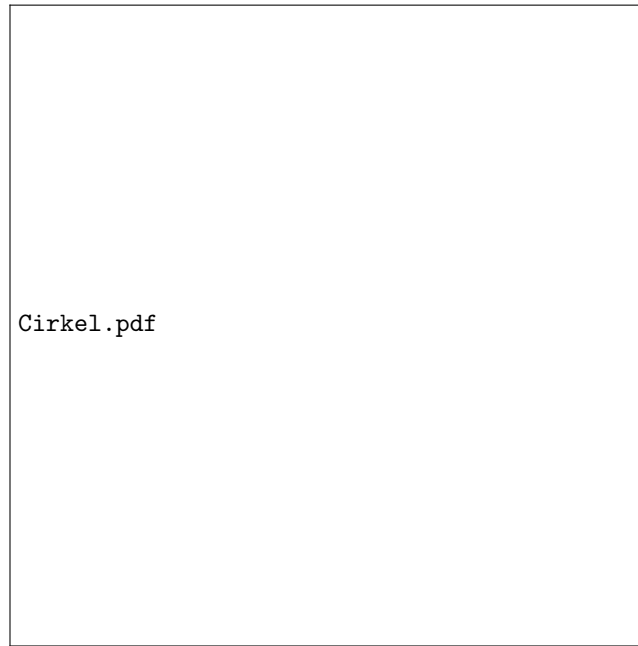


Figure 5: Unit circle where points P and Q are mirrored on the X-axes

$$\textit{Because } Xp = Xq, \cos(-\alpha) = \cos(\alpha) \textit{ applies} \quad (1)$$

$$\textit{Because } Yq = -Yp, \sin(-\alpha) = -\sin(\alpha) \textit{ applies} \quad (2)$$

This mirroring phenomenon is something that needs to be resolved. Mirroring of the angle means that the robot(s) that is being detected could be in two different direction relative to the interpreter. Figure 6 shows how the different robots determine the relative distance to each other.

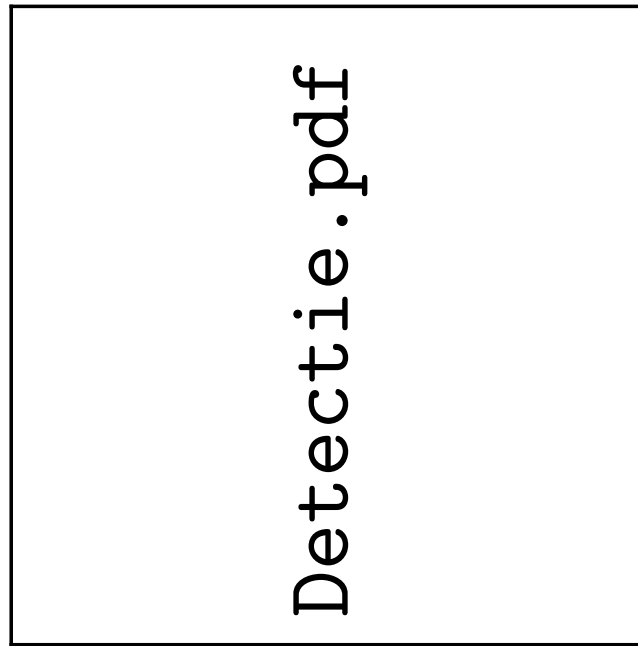


Figure 6: Relative location interpretation

In figure 6 a scenario is given where the relative distance between the different nodes (N1, N2 & N3) is known but their position within the circle is unknown. Because the relative direction (angle) from the interpreters (different nodes) needs to be determined. According to the goniometric equations the direction (angle) within a circle could be mirrored as shown in equation 1 and 2. This could lead to two different interpretations one being the reality as shown in figure 6 and one shown in figure 7 (where N1 is the interpreter).

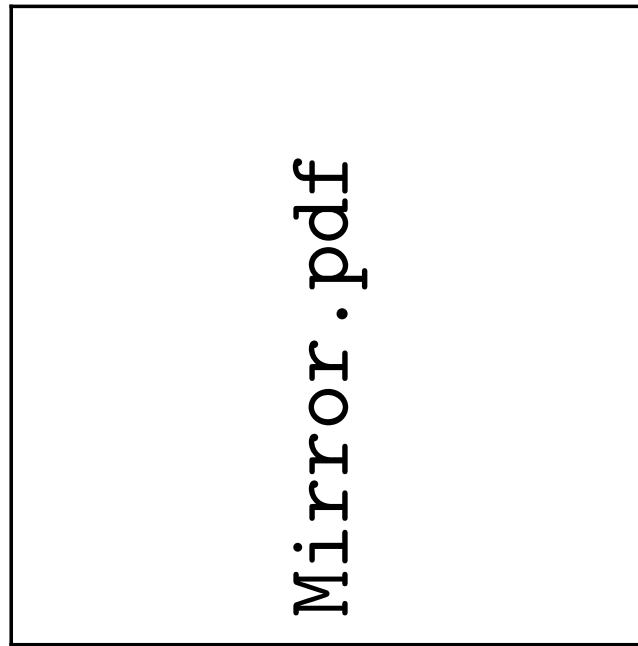


Figure 7: N1 Mirrored interpretation

For the robots to know in what direction the measured distance is applicable to, they can analyze the change in distance after a robot has moved a specific direction. In order to leverage the previous trilateration procedure requires coordinating the motion of the robots in a manner that gives every robot a chance to move and ensures that when a robot is moving its neighbours remain stationary. [1]

This method is very impractical because several robots need to abort their current task to move around. This method takes a lot of time to complete and therefore angular updates cannot be done very often. For previous research in this specific field, determining orientation with movement see.[?]

This can also be achieved by the use of stationary beacons. When a minimum of three beacons is placed on the field the location of the robots within this field can be measured, this is called triangulation. Triangulation is the process of determining the location of a point by measuring angles to it from known points. This is illustrated in figure 8.

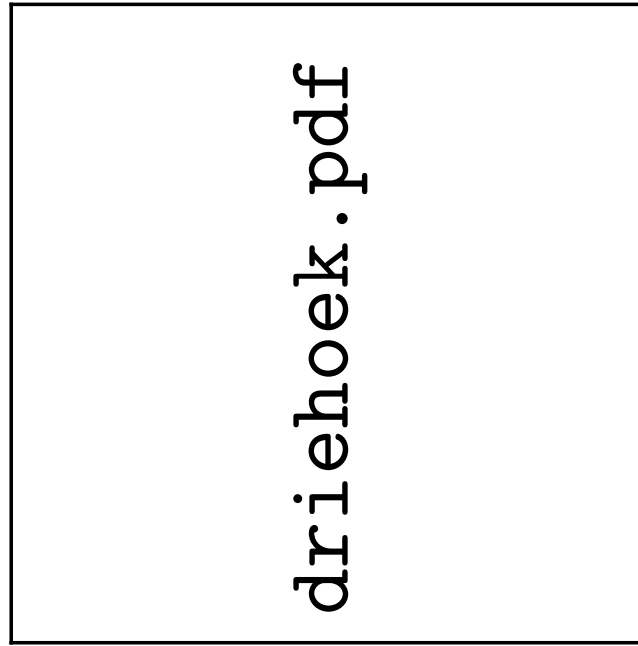


Figure 8: Triangulation using three stationary beacons

Using three stationary beacons the exact position of every node within this field can be calculated. This can be done by using the cosine rule shown in the equations below.

$$A^2 = B^2 + C^2 - 2 \cdot B \cdot C \cdot \cos(\alpha) \quad (3)$$

$$B^2 = C^2 + A^2 - 2 \cdot C \cdot A \cdot \cos(\beta) \quad (4)$$

$$C^2 = A^2 + B^2 - 2 \cdot A \cdot B \cdot \cos(\gamma) \quad (5)$$

With the cosine rule exact location of the different nodes can be determined. While using three stationary beacons mirroring cannot occur any more. Because if one beacon measures a distance to a node that is greater than the mutual-distance to another beacon the node is no longer located within the triangulation field.



Figure 9: Determining the angle with three stationary beacons

In figure 9 the distances from the beacons to node 1 (N1) are being determined. With these distances the different angles can be calculated this is done in the equation below.

$$A^2 = B^2 + C^2 - 2 \cdot B \cdot C \cdot \cos(\alpha) \quad (6)$$

$$\cos(\alpha) = \frac{(B^2 + C^2 - A^2)}{(2 \cdot B \cdot C)} \quad (7)$$

$$\frac{(12^2 + 8,26^2 - 7,41^2)}{(2 \cdot 12 \cdot 8,26)} = 0,7935 \quad (8)$$

$$\cos^{-1}(0,7935) = 37,48^\circ \quad (9)$$

A more efficient solution is to retrieve the relative orientation opposite to each other. In combination with the relative distance every robot knows the exact relative position of all the other robots in range of the network.



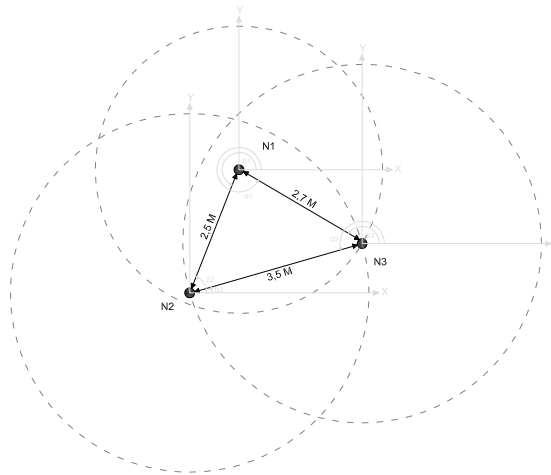


Figure 10: relative localization system using relative distance and orientation

As seen in figure 10 each node has a common reference direction (X,Y), the direction relative to this reference is called the relative orientation of the robot and can be visualized as shown in figure 10. With the known distances, the mapping of the swarm can be completed.

## 3.2 Relative Distance

Relative distance can be measured using various methods. The most common methods are "*Received Signal Strength Indication*" and "*Time Of Flight*" abbreviated as RSSI and TOF.

### 3.2.1 Received Signal Strength

A method to determine distance can be implemented using Received Signal Strength abbreviated as "RSSI". Typically RSSI is a measure of dBm, which is ten times the logarithm of the ratio of the power (P) at the receiving end and the reference power (Pref). Power at the receiving end is inversely proportional to the square of distance.[3] Hence RSSI could potentially be used as an indicator of the distance at which the sending unit (robot) is located from the receiving unit.[3]

RSSI is defined as ten times the logarithm of the ratio of power of the received signal and a reference power (for example 1mW). It is known that power dissipates from its source as it moves further out. As mentioned earlier the relationship between power and distance is that power is inversely proportional to the square of the distance travelled. [3]

In theory it seems as a suitable implementation for the purpose, determining relative distance between different units. However various test results have shown that RSSI is only reliable under certain circumstances. Test results have shown that if the orientation of the sending or receiving unit changes it becomes very unreliable. The graphs below show the reliability of RSSI when the direction is being altered.[3]

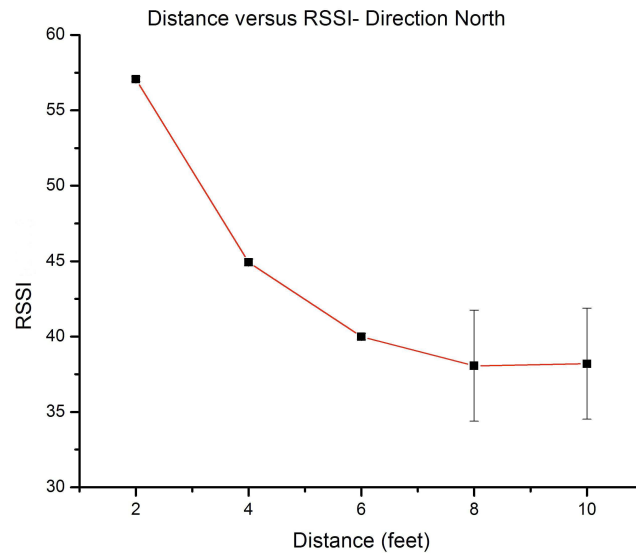


Figure 11: Distance versus RSSI plot showing inverse non linear relationship.[3]

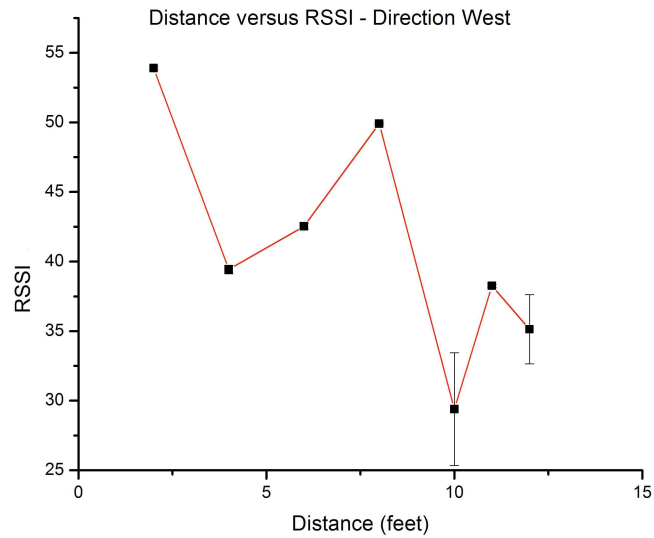


Figure 12: Distance versus RSSI plot showing lack of reliability when tested in a different direction.[3]

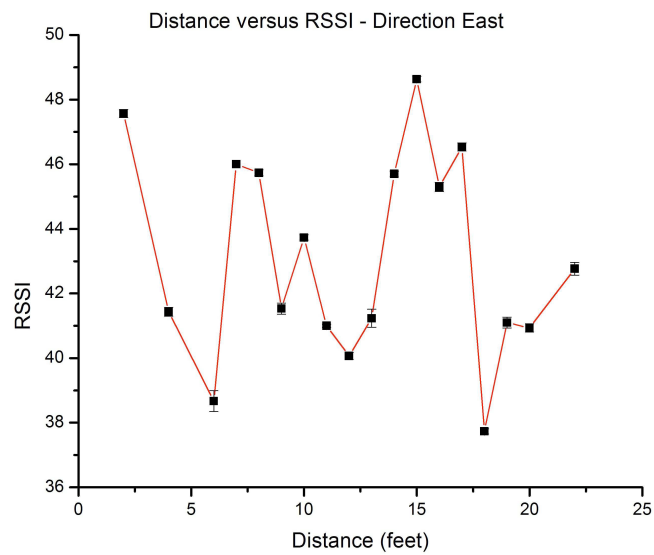


Figure 13: Distance versus RSSI plot showing lack of reliability when tested in a different direction.[3]

Even though RSSI was very promising in many studies, through multi directional experiments it became clear that even under ideal conditions with weather and interference controlled, the RSSI data could not be relied upon. At times the data was correct showing the expected inverse square relation with distance and at other times it didn't.[3]

### 3.2.2 Time Of Flight

Another common method to determine relative distance is Time Of Flight abbreviated as "TOF". Time Of Flight describes a method that measures the time that an particular wave takes to travel a distance through a medium. The wave can be acoustic, electromagnetic and light.

For relative distance measurements Time Of Flight is often combined with Time Difference Over Arrival abbreviated as "TDOA". Combining these methods enables relative distance measurements to specific units/nodes. TDOA measures the time difference between the transmission and arrival of a wave. With the velocity of propagation times the travelled time the relative distance between two or more nodes can be determined. However TOF also features certain limitations being clock synchronization, noise, sampling and multipath channel effects.[4]

*Clock Synchronization* TOF ranging systems need to estimate the time of transmission and arrival and comparing this with a common time reference.[4] If the different time references are not perfectly synchronized a time offset error occurs. With the faster the propagation speed, the greater the error in distance.

*Sampling* Using Light or electromagnetic waves to determine the relative distance can add several limitation. Due to the high velocity propagation (being nearly equal to equal of the velocity propagation of light) sampling this signal requires very high clock rates to do so. It would take a 15GHz clock to achieve one centimetre resolution.[5]

in the 1960's, Tektronix invented a sampling method with an oscilloscope that could measure signals in the GHz range without using high speed clocks. This method is called Sequential Equivalent Time Sampling abbreviated as "SETS". The SETS acquires one sample per trigger see figure 14. When a trigger is detected a sample is taken after a very short, but well defined delay. When the next trigger occurs a small time increment "Delta T" is added to this delay and the digitizer takes another sample. This process is repeated many times with "Delta T" added to each previous acquisition, until the time windows is filled.[6] In figure 15 another example of an sinusoidal signal is displayed using the SETS method.

Using the SETS method enables the use of waveform with an velocity propagation nearly equal to equal to the speed of light. However implementing a advanced SETS circuit is required to reach accuracy of less than 1 meter indoors.[4] However using waveforms with a high velocity propagation enables faster data transfers and a larger range.[4]

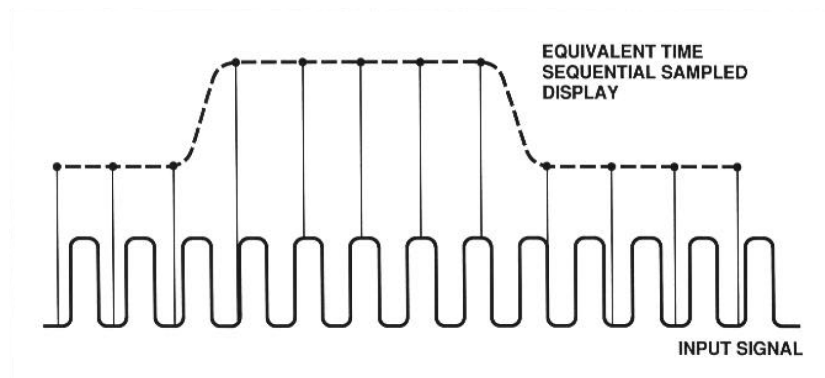


Figure 14: In sequential equivalent time sampling a single sample is taken for each recognized trigger after a delay which is incremented after each cycle.[6]

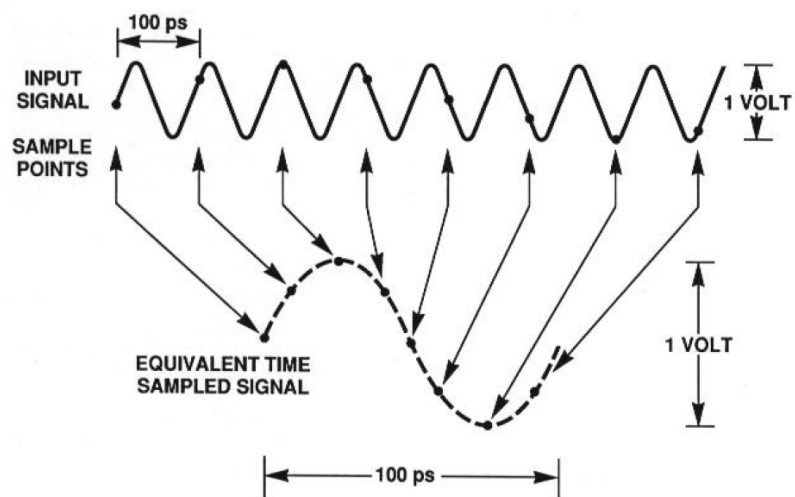


Figure 15: Example equivalent time sampled signal.[6]

### 3.3 Conclusion

Different studies have shown that Received Signal Strength can theoretically measure distance accurately. However RSSI seems to be very unreliable in many circumstances which leads to not being viable for distance measurements with moving robots.

Time Of Flight has been very promising, however using electromagnetic signals (For example RF, WI-FI or Bluetooth) the acceptable accuracy starts at from a minimum range of 1-2 meters which leads to unreliable to no useful information at all in the range of 1-2 meters. This minimum range could be narrowed using an Sequential Equivalent Time Sampling Circuit, however such a circuit is complex to develop. However using waveforms with a lower velocity propagation can lead to accurate close range measurements. In this case Time Of Flight suits better for the required use, being more accurate and reliable see figure 16.[4]

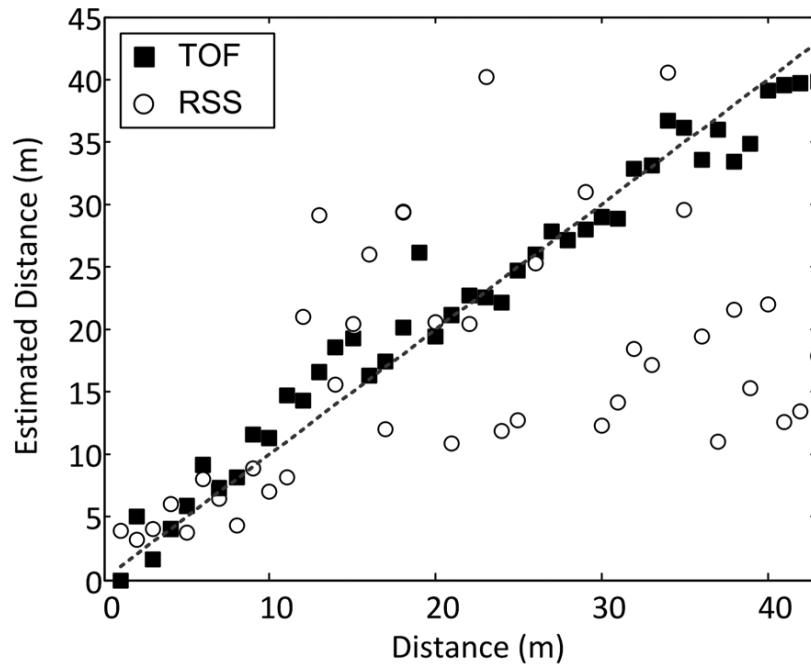


Figure 16: Accuracy RSSI vs TOF.[4]

Determining relative distance does not lead to localization to do so a angle is required to be continued...

## 4 Propulsion, Actuators and Effectors

In this chapter the sub-questions about the propulsion, actuators and effectors are researched and answered. First of all, the possible types of propulsion will be researched. Then the various kinds of actuators with a suitable effector will be considered. Depending on the situation different types of propulsion is used. For example a robot that operates in an environment with water would not be implemented with DC motor drivers and tyres. The robot build in this research should be able to operate on Mars in the future. Therefore the robot cant get stuck easily since that will cost millions, since the robot cannot complete his mission. Since swarming and not propulsion is the main goal of this research a simple way of locomotion would suffice. However there should be the possibility to implement a different kind of propulsion in the future, so that the robot can be used on Mars.

### 4.0.1 Effectors

The way a robot moves is determined by its effector with his corresponding actuator. Possible types of effectors are:

- Wheels
- Leg(s)
- Catepillar tracks
- Air cushion
- Propeller
- (Superconductor)

Each of these effectors have pro's and con's in different situations. Tyres are easily implemented, but there are various ways to implement these. The amount of tyres used will affect the handling of the robot. With a minimum two tyres, movement can already be realized. The downside is that driving with two wheels is not stable, only with algorithms that can balance the robot stable movement can be accomplished. It would be easier to implement more than two wheels so that the robot is more stable and movement is easier.

Elke van deze effectoren hebben voor en nadelen in verschillende situaties. Wielen zijn eenvoudig te implementeren, er kunnen verschillende aantallen wielen worden aangebracht. Het aantal wielen heeft wel invloed op het rijgedrag van de robot. Met twee wielen kan er worden gereden, mits de robot in balans wordt gehouden. Dit kan worden gedaan door middel van een zwenkwiel, of door een regelsysteem te bedenken dat de twee wielen de robot in balans houden. Dit laatste kost veel energie en is niet eenvoudig. Met drie wielen staat de robot al een stuk stabiel. Het nadeel hiervan is dat sturen soms moeizaam kan zijn. Met vier wielen is de robot nog stabiel en kan er op verschillende manieren worden gestuurd. De wielen kunnen tegengesteld worden aangedreven waardoor de robot draait. Dit kost wel meer energie dan nodig is aangezien er extra wrijving ontstaat. Een optie is de voorste wielen te laten draaien, waardoor de robot in de richting van de wielen zal voortbewegen. Zoals bij een auto het geval is. Dit is weer lastiger te implementeren aangezien er een soort van stuurmechanisme moet worden bedacht.

#### **4.0.2 Actuatoren**

De keuze van de actuator hangt sterk samen met de effectoren.

#### **4.0.3 Aandrijving**



## 5 Swarm Communication

There are several requirements to realise a robot swarm. Each member of the population needs to communicate with the rest of the swarm. To give the members full freedom of movement the communication also needs to be wireless. It is important that the population size of the swarm is dynamic, because of the scalability of the swarm. As mentioned earlier a swarm does not have a master. So each member needs to be capable to maintain the network. These networks also must be able to reroute around nodes that have been lost.

To create an network with those characteristics a (wireless) mesh network ((W)MN) is a possible solution. A mesh network relies on communication nodes in which data is distributed. All of the communication nodes cooperate in the distribution of data in the network. Each node of the network is connected to one or more nearby in range node(s) and forwards data on behalf of the other nodes. [7] A example of an mesh network is illustrated in figure 17. To ensure a reliable network, self-organization and topology control algorithms are needed.[8] Mesh networks can use self-healing algorithms such as Shortest Path Bridging (SPB) to restore and reroute the connection around broken nodes/clients. Shortest path bridging calculates the most favorable connections to the clients with the lowest path costs. [9] Other usefull algorithms are listed in [10]. The most efficient algorithm for path bridging depends on the situation where the clients are operating in which is described in [10].

A wireless mesh network (WMN) is a dynamically self-organized and configured network. [8] Each node or client in the network is able to create an ad hoc network to maintain the mesh connectivity, this is called client WMN's. These networks are able to reroute around nodes that have been lost. On large scale projects wireless mesh networks are costefficient because of the saving of wired connections.[7]

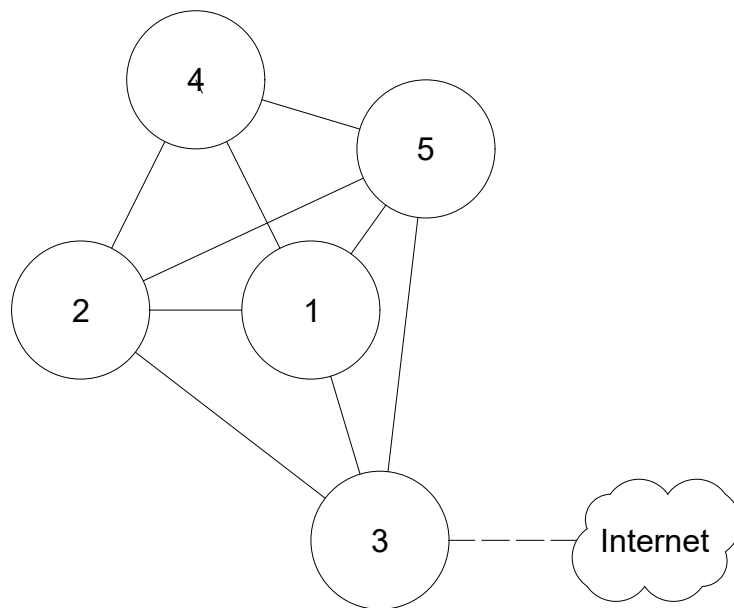


Figure 17: An example of a wireless mesh network with 5 clients and an optional internet connection. The connections made in this example are random.

There exist several techniques to accomplish this method Examples of possible protocols:

## 5.1 Proximity sensor

For close range obstacle avoidance (0m - 0,3m, the robots will need some kind of proximity sensing technique. The sensor needs to differentiate between different "treats" for the robot to function properly. For example; when the sensor detects a small hill, which the robot can move over it should not trigger the robot to move away from it. But when it detects a big obstacle it should trigger the robot to move away. This is important because the robots will eventually be expected to move across rough terrain (like Mars). In the next paragraphs some different proximity sensor techniques will be discussed. Price is also an important factor while comparing these techniques. Multiple robots must be made on a tight budget, so costs should be cut where possible.

### 5.1.1 Ultrasonic sensor

Ultrasonic sensors are widely used in robotics for proximity sensing. An ultrasonic sensor uses the time of flight (TOF) to operate. It sends out an ultrasonic sound wave and measures the time it takes for the sound to be reflected back. The speed of the sound is known so the distance can be calculated. The fact that this method uses sound is a big advantage when measuring small distances. Sound has a velocity of propagation of around 340 m/s. Most other methods use signals that travel with the speed of light ( $3 \cdot 10^8$ ). The problem with the high velocity of propagation especially with measuring distances shorter than one meter is that the TOF will be in terms of a few nanoseconds. The hardware needed to process signals with this kind of speed will require more complex circuit design and ultimately cost more. The ultrasonic signal that is received after reflected from an object holds information about the shape and size of the object. This can be used to discriminate between different obstacles[?]. Because there's a different atmosphere on Mars ultrasonic sound gets heavily dappened [11]. This means that this method won't be usable on Mars itself. A pro of the ultrasonic sensor is the low cost. A sensor with specified for short distances costs between 2€ and 10€.

### 5.1.2 Lidar sensor

Light detection and ranging (lidar) works with the same principle as the ultrasonic sensor. It uses time of flight to determine the distance from an object. Instead of sound it uses the light of a laser as its medium. Lidar can be used for various applications like: 3D modeling of an environment, weather forecast and distance measurements [?]. Lidar is a proven technology on Mars, it has been used on the Phoenix mission to measure clouds and atmospheric dust [12]. A lidar suitable for the purpose of proximity sensing would be the Lidar lite v2. This is a proximity sensor designed for robotics. It has a range 0-40m and an accuracy of  $\pm 0,025$ m. Being the cheapest lidar sensor on the market it still costs 100€. Object discrimination with the Lidar lite is not possible due to the limited information. More complex lidar systems with a moving laser could be able to discriminate and detect objects really precise, but this requires really complex systems.

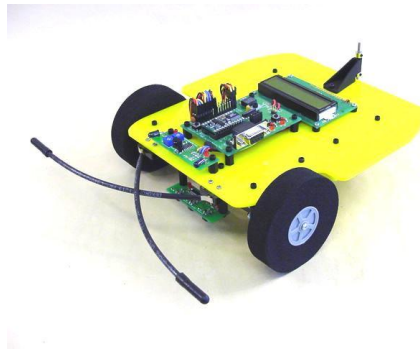


Figure 18: Robot met mechanische proximity sensor

### 5.1.3 Mechanische sensor

A mechanical sensor will consist of two important parts. The actuator, this could be a simple switch sensitive to a certain pressure. And the so called "arm" or "bumper", this is the part that makes contact with the obstacle's surface and works the mechanical force on the actuator. The beauty of this system is it's simplicity. There's no fast signals with different frequencies to be processed, just a simple on or of signal is all that needs to be observed. Due this limited input object discrimination won't be possible. Also robots bumping into object and each other is not a preferable situation. The robots might get stuck or break down.

### 5.1.4 Sensor choice for the robot

Now the question is which of the different sensor techniques proposed, is the best to implement on the swarm robots. While answering this question the following points will be taken into account:

- The sensor must be easy to reproduce because multiple robots must be made from the swarm.
- The budget is limited, therefore the sensor price should be cheap compared to the total price of one robot.
- Energy usage should be as low as possible to increase the operation time per robot.
- For budget reasons there has been chosen to not take into account if the sensor would work on Mars.
- It must be possible to discriminate between different obstacles.

Looking at these points it becomes clear that the lidar sensor wouldn't be the best choice. It doesn't really have any outstanding pros compared to the other two techniques and it's much more expensive. . As talked about before the mechanical sensor has a few drawbacks compared to the ultrasonic sensor. For these reasons the ultrasonic sensor seems the best choice. The only issue left to solve is how to sense the difference between obstacles which the robot has

to avoid, and things the robot doesn't have to avoid, but still might be detected by the sensor. This will be discussed in the next paragraph.

### 5.1.5 Obstacle discrimination with an ultrasonic sensor

An ultrasonic sensor is usually used to detect if there is some obstacle in the way. In this case there is made no difference made between a obstacle that the robot can move over, or a obstacle that should be avoided. For example: a small slope or hill might be detected as an obstacle, but in truth the robot can move over it. The robots developed for this program will eventually be able to cross harsh terrain. In this case the proximity sensor will need to discriminate between these situations. Working with ultrasonic waves there are multiple domains to work with. These are time, frequency and amplitude. Time is used to determine the distance between the object and the sensor. The change in frequency and/or amplitude contains information about the shape of the object[?]see figure 19. In the paper "Object recognition with ultrasonic sensor" is concluded that looking at the amplitude over time contains enough information to discriminate between object[?]. The example is given of an object with separated surfaces of 3,5 cm, the echo from the second surface will be delayed by 0,2 msec relative to the echo from the first surface. This delay can be easily detected with a microprocessor. The reason looking at frequency isn't the first choice is because of the higher hardware requirements it would require to properly sample the ultrasonic signal. Looking at amplitude and time does not require the signal to be sampled at the Nyquist rate so lower clock rate microprocessors can be used to do the processing.

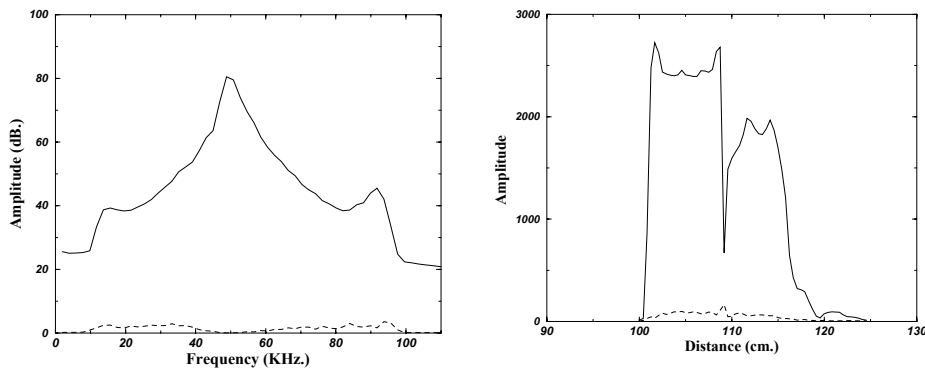


Figure 19: The left graph shows the average PSD from 50 echos generated by a plastic bottle from a distance of 100cm, the average envelope is generated using 50 echos of the same bottle

[?]

## 6 Bibliografie

### References

- [1] *Distributed Range-Based Relative Localization of Robot Swarms*, Harvard University, 1 2014.
- [2] *A New Three Object Triangulation Algorithm Based on the Power Center of Three Circles*, INTEL SIG Laboratory, Montefiore Institute, University of Liege, Belgium, 1 2011.
- [3] *Is RSSI a Reliable Parameter in Sensor Localization Algorithms – An Experimental Study*, State University of New York at Buffalo, 1 2010.
- [4] *Radio Frequency Time-of-Flight Distance Measurement for Low-Cost Wireless Sensor Localization*, University of California And San Diego, 3 2011.
- [5] *Open source, TOF laser rangefinder*, Arduino Forums, 1 2014, <https://forum.arduino.cc/index.php?topic=213774.0>.
- [6] *Sampling Oscilloscope Techniques*, Tektronix, 1 1960.
- [7] Wikipedia. Mesh networking. [Online]. Available: [https://en.wikipedia.org/wiki/Mesh\\_networking](https://en.wikipedia.org/wiki/Mesh_networking)
- [8] *A Survey on Wireless Mesh Networks*, Georgia institute of technology, 9 2005.
- [9] *Shortest Path Bridging*, M. Seaman, 1 2005.
- [10] *Position-Based Routing in Ad Hoc Networks*, University of Ottawa and Universidad Nacional Autonoma de Mexico, 6 2002.
- [11] A. D. Hanford. Computer simulations of propagation of sound on mars. [Online]. Available: <http://acoustics.org/pressroom/httpdocs/151st/Hanford.html>
- [12] *Lidar on the Phoenix mission to Mars*, James Whiteway, 6 2008, vertrouwelijk.