

Zelfkalibratie-algoritme voor grootschalige UWB-systemen gebaseerd op Simulated Annealing

Joppe Smeets

Studentennummer: 01700150

Promotoren: prof. dr. ir. Eli De Poorter, prof. dr. ir. Jeroen Hoebeke

Begeleiders: Matteo Ridolfi, ir. Jaron Fontaine, ir. Ben Van Herbruggen

Masterproef ingediend tot het behalen van de academische graad van
Master of Science in de industriële wetenschappen: informatica

Academiejaar 2020-2021

”De auteur geeft de toelating deze masterproef voor consultatie beschikbaar te stellen en delen van de masterproef te kopiëren voor persoonlijk gebruik. Elk ander gebruik valt onder de bepalingen van het auteursrecht, in het bijzonder met betrekking tot de verplichting de bron uitdrukkelijk te vermelden bij het aanhalen van resultaten uit deze masterproef.”

7 juni 2021

Dankwoord

Ik zou graag mijn promotoren prof. dr. ir. Eli De Poorter en prof. dr. ir. Jeroen Hoebeke bedanken voor de mogelijkheid om dit onderwerp als thesis te hebben. Daarnaast wil ik ook graag mijn begeleiders Matteo Ridolfi, ir. Ben Van Herbruggen en ir. Jaron Fontaine bedanken voor de hulp en de feedback tijdens deze thesis. Het is jammer dat we elkaar omwille van de coronapandemie nauwelijks in het echt gezien hebben, maar ik ben wel blij met de begeleiding en het verloop van de communicatie. Ik ben ook zeer tevreden met de vrijheid die ik gekregen heb. Dit was goed in balans met de duidelijke richtlijnen en aanbevelingen die mij geholpen hebben bij de uitwerking van deze thesis.

Zelfkalibratie-algoritme voor grootschalige UWB-systemen gebaseerd op Simulated Annealing

Promotoren: prof. dr. ir. Eli De Poorter, prof. dr. ir. Jeroen Hoebeke
Begeleiders: Matteo Ridolfi, ir. Jaron Fontaine, ir. Ben Van Herbruggen

Masterproef ingediend tot het behalen van de academische graad van:
Master of Science in de industriële wetenschappen: informatica

Faculteit Ingenieurswetenschappen en Architectuur

Academiejaar 2020-2021

In de laatste decennia is de belangstelling voor positiebepalingssystemen sterk gegroeid. Ultra-wideband (UWB) is een van de populairste technologieën wanneer een hoge nauwkeurigheid vereist is bij *indoor* toepassingen. Deze nauwkeurigheid is te danken aan de unieke signaaleigenschappen die het ook mogelijk maken om de afstand te bepalen in *non-line-of-sight* omgevingen. De bestaande lokatiesystemen vereisen doorgaans dat de posities van de anchors op voorhand bekend zijn. Dit vereist echter dure meetinfrastructuur en is gevoelig voor fouten. Dit tijdrovend proces leidt tot hoge installatiekosten en beperkt de toepassing in verschillende situaties. Zo is het nauwelijks bruikbaar in situaties waar er onvoldoende tijd is, zoals bij reddingoperaties. Om deze problemen op te lossen werden reeds verschillende zelfkalibratie-algoritmes ontworpen die automatisch de positie van de anchors kunnen bepalen. Tot nu toe werden deze algoritmes voornamelijk getest in line-of-sight omgevingen met een beperkt aantal anchors. Daarom wordt er in deze thesis een eerste stap gezet in de richting van zelfkalibratie-algoritmes voor grootschalige systemen.

Het experimenteel gevalideerde algoritme is gebaseerd op bestaande *simulated annealing localization* algoritmes en werd getest in een uitdagende NLOS-omgeving met een oppervlakte van ongeveer 1000 m^2 . Dit algoritme behoort tot de categorie van multi-hop algoritmes waardoor er meer vrijheid is bij de plaatsing van de anchors. In tegenstelling tot de single-hop algoritmes moet een anchor zich niet noodzakelijk in het bereik van minstens 4 gelokaliseerde anchors bevinden om zijn positie te bepalen. Daarnaast is het algoritme goed bestand tegen de voortplanting van fouten. Dit leidt tot een gemiddelde positiebepalingsfout van slechts 50 cm afhankelijk van de anchorconfiguratie.

Sleutelwoorden: UWB, Positiebepaling, Zelfkalibratie, Simulated Annealing

Simulated Annealing based self-calibration algorithm for large-scale UWB systems

Joppe Smeets

Supervisors: Prof. dr. ir. Eli De Poorter, Prof. dr. ir. Jeroen Hoebeke
Counsellors: Matteo Ridolfi, ir. Jaron Fontaine, ir. Ben Van Herbruggen

Abstract—In the last decades, the interest in localization systems has increased notably. Ultra-wideband (UWB) is one of the most popular technologies when high accuracy is required due to its advantageous signal properties. The existing solutions usually require that the positions of the anchors are known in advance. However, obtaining the anchor positions manually is a time-consuming task that requires expensive measurement infrastructure such as laser meters. To solve these problems, several self-calibration algorithms have already been developed that can automatically determine the position of the anchors. Until now, these algorithms were mainly tested in line-of-sight (LOS) conditions with only a limited number of anchors. Therefore, in this thesis a self-calibration algorithm is proposed that is suitable for large-scale systems in NLOS conditions. The algorithm is based on simulated annealing localization and is also suitable in multi-hop environments where one anchor is not necessarily located within the transmission range of all other anchors. During the experimental validation in a test environment of 1000 m², a positioning error of 50 cm was achieved depending on the anchor configuration.

Keywords—UWB, localization, self-calibration, multi-hop, simulated annealing

I. INTRODUCTION

The purpose of indoor localization systems is to determine the location of objects or people in places where GPS and other localization techniques do not provide sufficient accuracy. Indoor localization can then be used to, for example, help passengers in airports, guide visitors around a museum or provide drones with accurate locations. Positioning systems based on satellites are often not useful in buildings because the satellite signals have difficulty penetrating walls and ceilings. Ultra-wideband (UWB) is better suited in NLOS conditions thanks to its unique signal properties. The disadvantage of ultra-wideband and other RF based distance determination systems is that they usually require the positions of the anchors to be known in advance. In this thesis, a self-calibration algorithm is proposed that can automatically determine the positions of the anchors. The purpose of this self-calibration procedure is to minimize installation costs and save time while deploying the UWB infrastructure. As a result, indoor localization systems can also be convenient in situations where only a temporary solution is needed, such as in rescue operations.

The remainder of this extended abstract is structured as follows: Section II provides an overview of related work on self-calibration and multi-hop localization algorithms. Next, in Section III, the simulated annealing based self-calibration algorithm is presented. In Section IV the test environment and the approach will be discussed. This is followed by a discussion of the results in Section V. The abstract is concluded in Section VI, where the conclusions and directions for future research are proposed.

II. RELATED WORK

This thesis finds itself at the intersection of two specific research fields: self-calibration of the anchors (i) and multi-hop localization (ii). Most of the methods discussed in this thesis are closely related to proposals in existing literature. The main contribution of this thesis lies in the combination of the existing algorithms from both categories to obtain a robust algorithm that is suitable for self-calibration of the anchors in challenging NLOS conditions and multi-hop environments. In this section, the basic concepts and limitations of existing self-calibration and multi-hop localization algorithms are discussed.

A. Self-Calibration

The existing self-calibration algorithms usually start with defining the coordinate system. Usually 1 anchor is indicated as origin. In addition, 3 other anchors are required to be located along the positive x-, y- and z-axis respectively. The disadvantage of this method is that the accuracy of the system is highly dependent on the accuracy of the distance measurements between the 4 anchors used to define the coordinate system (limitation 1). Small errors in these distance measurements can lead to an incorrect determination of the coordinate system.

Table I provides an overview of existing self-calibration methods. This overview is based on the self-calibration survey from Ridolfi et al. [1]. These methods were mostly evaluated in a small LOS environment where the number of anchors is rather limited. Usually, single-hop techniques are used to determine the positions of the remaining anchors (which were not used when defining the coordinate system) in small-scale environments.

TABLE I: Overview of existing self-calibration methods

Solution	Anchors	Hop	Size	Accuracy
Bastone et al. [2]	6	Single	20 x 20 m ²	15 cm
Hamer et al. [3]	8	Single	6 x 7 x 3.5 m ³	9.7 cm
Shi et al. [4]	5	Single	20 x 20 x 20 m ³	15 cm
De Preter et al. [5]	4	Single	8 x 7 m ²	10 cm
Yu et al. [6]	4	Single	12 x 8 m ²	10 cm
Almanasa et al. [7]	4	Single	144 m ²	135 cm
Ridolfi et al. [8]	4	Single	10 x 4 m ²	2.4 cm
Our solution	15	Multi	40 x 25 x 2.6 m³	50 cm

In single-hop algorithms, it is assumed that all nodes in the network can directly communicate with at least 4 nodes whose position is known. However, for larger systems, the chance that an anchor with an unknown position will be within the transmission range of 4 localized anchor reduces (limitation 2). These limitations can be mitigated by using multi-hop algorithms. This type of algorithms is discussed in the next section.

B. Multi-Hop Localization

Table 2 provides an overview of different range-based multi-hop localization algorithms that serve as the basis for other multi-hop algorithms. This overview is based on the work of Mao et al. [9].

TABLE II: Overview of multi-hop localization algorithms

Solution	Topology	Algorithm
Ji et al. [10]	Centralized	multidimensional scaling (MDS)
Biswas et al. [11]	Centralized	semi-definite programming (SDP)
Kannan et al. [12]	Centralized	simulated annealing (SA)
Savarese et al. [13]	Distributed	DV-distance
Capkun et al. [14]	Distributed	local to global maps
Ihler et al. [15]	Distributed	particle filters

These algorithms show fundamental differences from the single-hop localization algorithms [16]. Single-hop algorithms assume that all nodes in the network can communicate directly with a set of nodes whose position is known. The goal of multi-hop algorithms is to overcome this limitation. The number of studies comparing different multi-hop algorithms is rather limited. This makes it difficult to select the most suitable algorithm from this overview. Ultimately, the simulated annealing algorithm was chosen because of the following favorable properties:

- the resistance to error accumulation due to errors in the distance and position determination;
- the ability to escape from a local optimum, which may allow the algorithm to find a better approximation of the global optimum;
- the freedom in anchor selection.

The other algorithms can be seen as valid competitors with similar properties. Further research is required to find out which algorithm has the best performance in certain circumstances. In the next section, the selected simulated annealing localization algorithm is discussed.

III. SIMULATED ANNEALING

Simulated Annealing (SA) is a stochastic algorithm that explores its environment from an initial solution in order to find an approximation of the global optimum of a given function. During the exploration, the cost function is used as an indicator of the quality of the solution. The name of this optimization algorithm comes from annealing in metallurgy: a metal is heated and then cooled in a controlled manner with the aim of increasing the number of crystals and thus making the metal stronger. At high temperatures, the molecules can move freely and major changes occur in the structure of the metal. This temperature decreases over time along with the probability that a molecule will move.

The basic operation of the algorithm is shown in pseudocode in Algorithm 1. This is only one possible implementation as there are alternatives for several steps [17]. The algorithm requires that the locations of at least 3 nodes are known in advance. In the initial phase of the algorithm, all anchors with an unknown location are initialized with an arbitrary location that is located within the specified dimensions of the setup (line 1). This is a solution that meets all the requirements but will rarely be the best solution. Then the temperature (line 2) and the displacement distance (line 3) are set. From this initial solution, a new candidate solution (line 6) is generated by moving a random anchor whose location is not yet known in any direction over a specified distance. This candidate solution is evaluated using the acceptance criteria (line 7).

The acceptance scheme is shown in pseudocode in Algorithm 2. Better solutions are always accepted. There is also a chance that a less good solution will be accepted, but that chance decreases as the temperature of the system decreases. This temperature changes according to a predefined cooling schedule (line 11). After each

Algorithm 1: Simulated annealing localization

Input: distance measurements, coordinates of known anchors, room dimensions

Output: estimated position for each anchor

```

1 Create initial solution  $S$ 
2 Set temperature  $T$ 
3 Set move distance  $d$ 
4 while StopConditions not met do
5   for each anchor with an unknown position do
6     // move the anchor of solution  $S^*$  by a
7     // distance  $d$  in an arbitrary direction
8     Create new solution  $S'$ 
9     if AcceptanceConditions met then
10      // depending on the difference in
11      // cost and the temperature  $T$ 
12      Accept  $S'$ 
13    else
14      Reject  $S'$ 
15  Update  $T$  // cooling schedule
16  Update  $d$ 
17 Print solution  $S^*$ 

```

iteration of the algorithm, the stop conditions (line 4) are evaluated. If these conditions are met, the algorithm stops and the best candidate solution is returned as a result (line 13).

Algorithm 2: Acceptance criteria

Input: temperature T , CF_{new} , CF_{old}

Output: True/False

```

1  $\Delta(CF) = CF_{new} - CF_{old}$ 
2 if  $\Delta(CF) \leq 0$  then
3   // Downhill move  $\Rightarrow$  accept always
4   Return True
5 else if random number from uniform distribution (0,1)
6    $\leq e^{-\Delta(CF)/T}$  then
7   // Uphill move  $\Rightarrow$  accept with probability
8   Return True
9 else
10  // Reject candidate
11  Return False

```

This method uses a cost function as an indicator of the quality of the solution. The equation of the cost function (CF) is shown in Equation 1.

$$CF = \sum_{i=1}^N \sum_{j \in N_i} (d'_{ij} - d_{ij})^2 \quad (1)$$

$$d'_{ij} = \sqrt{(x'_i - x'_{ij})^2 + (y'_i - y'_{ij})^2 + (z'_i - z'_{ij})^2}$$

N : total number of unknown anchors

N_i : collection of the neighbors of anchor i

d'_{ij} : estimated distance between anchor i and neighbor j

d_{ij} : measured distance between anchor i and neighbor j

(x'_i, y'_i, z'_i) : approximate coordinates of anchor i

$(x'_{ij}, y'_{ij}, z'_{ij})$: approx. coordinates of neighbor j from anchor i

Anchors are neighbors when they are in each other's transmission range. Note that the set N_i can therefore contain both anchors whose location is known in advance, as well as anchors whose location has yet to be determined. The cost is recalculated for each new candidate solution. This solution will always be accepted when the difference in cost between the new and the old solution is less than or equal to zero (Equation 2).

$$\Delta(CF) = CF_{new} - CF_{old} \quad (2)$$

If $\Delta(CF) \geq 0$, then the Metropolis condition determines whether this less favorable solution will be accepted. This depends on the temperature, the associated cooling schedule and $\Delta(CF)$. Typically, the temperature and cooling schedule are determined empirically. The generic simulated annealing algorithm is discussed in more depth in [17], while [18] discusses in detail how it can be applied to localization.

IV. TESTENVIRONMENT AND APPROACH

The test environment consists of 15 anchors spread over an area of approximately $40 \times 25 \text{ m}^2$. This experimental setup is located in an office environment where walls and other obstacles create challenging NLOS conditions. The number of neighbors of an anchor varies from 3 to 12. An anchor has an average of 7 neighbors of which there is usually only 1 neighbor that is in the LOS of the anchor. Figure 1 shows the layout of the anchors in the test environment. The anchors are fixed to the ceiling in different offices, meeting rooms and corridors on floor 9 of the iGent building. Each anchor consists of an Intel NUC equipped with an UWB chip whose hardware is discussed in detail in [19]. Figure 2 gives an overview of the distribution of the absolute ranging error of the UWB measurements. For each pair of anchors, the median of approximately 500 distance

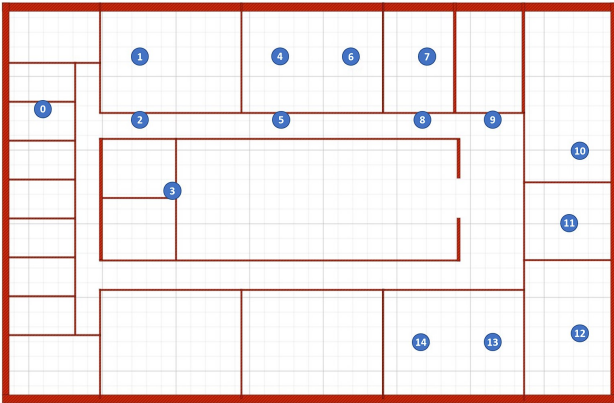


Fig. 1: Experiment setup with 15 anchors spread over an area of $40 \times 25 \text{ m}^2$

measurements was used as input to the algorithm. This resulted in a mean distance measurement error of 337 mm. In addition to the accuracy of the distance measurements, the placement of the anchors also has a significant influence on the accuracy of the position determination. Therefore, a total of 30 different anchor configurations are used for the validation of the algorithm. There are 10 anchor configurations where the position of 3 anchors is known in

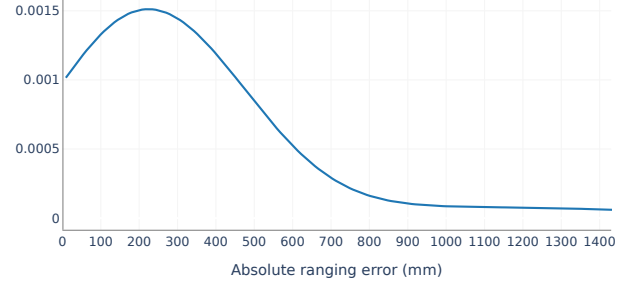


Fig. 2: Distribution of the absolute ranging error of the UWB measurements

advance. These configurations were extended with 1 and 2 additional randomly selected anchors. Each configuration uses the same distance measurements. The only difference is in the selection of the anchors that are known in advance. In addition, the z-coordinates of all anchors are also used as input for the algorithm, but this is not required. This choice was made because the z-coordinates of the anchors hardly differ from each other ($\Delta z = z_{max} - z_{min} < 10 \text{ cm}$).

V. RESULTS

In this section, the simulated annealing based self-calibration algorithm is evaluated. First, the influence of the number of known anchors on the mean absolute error (MAE) in position estimation is investigated. Then, the problems that may arise during the execution of the self-calibration algorithm are briefly discussed and illustrated.

A. Localization Error

After the basic functioning of the algorithm was verified using the laser measurements as ground truth, the algorithm could be applied to the UWB distance measurements. A total of 30 different anchor configurations were used for the validation of the algorithm, with the number of known anchors ranging from 3 to 5. Figure 3 shows the MAE in which the different anchor configurations were grouped by the number of anchors known in advance. Each group consists of 10 anchor configurations and 20 simulations were performed per configuration. For the configurations with 3 known anchors, the MAE

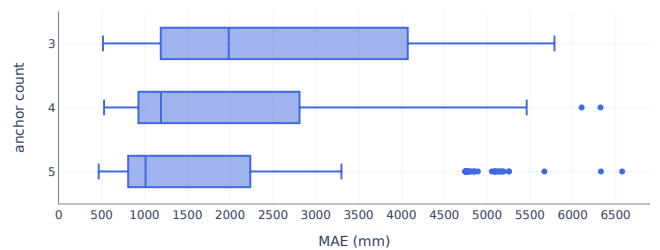


Fig. 3: Correlation between localization error and number of known anchors

is 1981 mm (median). This error decreases to 1010 mm as the number of known anchors increases. This decrease is less noticeable when comparing the minima between the different groups. The minima and medians are in line with expectations, but the spread and maxima are higher than expected. Possible causes for these problems are briefly discussed in the following sections.

B. Flip Ambiguities

The presumed main cause of the high spread and maxima is the presence of flip ambiguities. This type of problem is discussed extensively in [9] and often occurs when an anchor's neighbors are nearly collinear. Figure 4 shows the top view of the test setup with the actual and estimated locations represented by a rhombus and a star, respectively. The remarkable thing about these results is that the estimated positions of anchors 4, 6 and 7 appear to be reflected over their neighbors (2, 5 and 8). This is most likely due to the geometry of the anchors and the uncertainty in the position determination from the neighbors. Once the position of 1 of the neighbors is mirrored, this can cause an avalanche effect [18] which in turn causes the neighbors of a mirrored anchor to also be reflected over their neighbours.

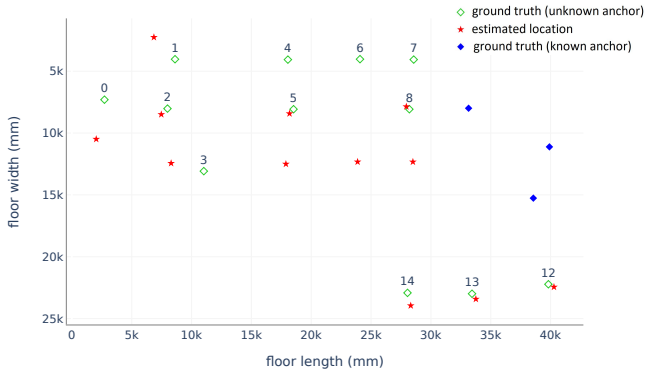


Fig. 4: Estimated anchor locations - flip ambiguities

C. Cost function

Another factor that has a major influence on the final result is the cost function. This function is used as an indicator of the quality of a solution. The goal of the simulated annealing algorithm is to find the global minimum of this function. Figure 5 shows the relationship between cost and MAE for 20 simulations. In general, a lower cost

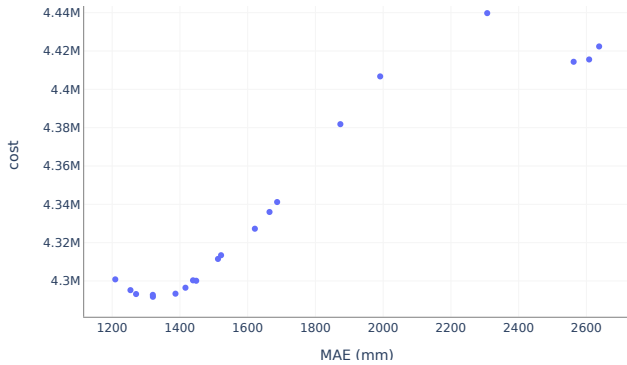


Fig. 5: Correlation between cost and localization error

indicates a better solution in terms of mean absolute error. However, there are some simulations that deviate from this and, despite a lower cost, still have a higher absolute error. Some possible causes of this are the presence of flip ambiguities and measurement errors. These can cause the cost function to be less representative, so that the estimated positions can deviate more from reality.

D. Parameter Initialization

Finally, there are also improvements in parameter initialization. The parameters in the simulated annealing algorithm play a major role in accepting and rejecting new solutions. It is important to accept enough less good solutions (higher cost) so that the algorithm can perform a thorough exploration of the environment. However, accepting too many bad solutions can cause the algorithm to never converge. It is therefore important to find a good balance so that the global optimum is approached as closely as possible.

In the literature there are only a few rules of thumb for setting these parameters appropriately [20]. For example, it is recommended to set the initial temperature so that the chance of accepting a less good solution (with a higher cost) is about 80% [12]. However, due to the random placement of the anchors in the initial phase of the algorithm, large differences can arise in the initial cost of the setup, making it difficult to find a suitable initial temperature. A possible solution for this is to set the parameters automatically depending on the cost achieved after a few short test runs.

VI. CONCLUSIONS

The simulated annealing-based self-calibration algorithm is robust and can automatically determine the position of the anchors in this challenging real-life environment. Despite an average ranging error of 34 cm and outliers up to 452 cm, the algorithm succeeds in limiting the error accumulation. This results in a MAE that varies from 50 cm to 250 cm depending on the anchor configuration.

One of the causes of a high MAE is the presence of flip ambiguities. Different flip ambiguity mitigation methods already exist, but these are less suitable in NLOS conditions because they are based on the transmission range of the nodes and this can vary greatly depending on the presence of obstacles. The presence of flip ambiguities can be reduced by executing the algorithm several times and then only continuing with the best intermediate result. However, it is important to accurately set the parameters of the simulated annealing algorithm. The cost function that determines the goodness of a solution manages to detect flip ambiguities. However, sometimes the algorithm gets stuck in a local minimum. This indicates that the parameters are not yet optimally set for the different configurations. An automatic configuration of the parameters could be a possible solution for this.

For future work it would be interesting to compare this algorithm with other self-calibration algorithms suitable for multi-hop environments. In addition, it is also possible to investigate how this algorithm can be further improved so that it is less computationally intensive and can be used on a large scale.

REFERENCES

- [1] M. Ridolfi, A. Kaya, R. Berkvens, M. Weyn, W. Joseph, and E. D. Poorter, "Self-calibration and collaborative localization for uwb positioning systems: A survey and future research directions," vol. 54, no. 4, 2021. [Online]. Available: <https://doi.org/10.1145/3448303>
- [2] K. Batstone, M. Oskarsson, and K. Åström, "Towards real-time time-of-arrival self-calibration using ultra-wideband anchors," *2017 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2017*, vol. 2017-Janua, pp. 1–8, 2017.
- [3] M. Hamer and R. Dandrea, "Self-Calibrating Ultra-Wideband Network Supporting Multi-Robot Localization," *IEEE Access*, vol. 6, pp. 22 292–22 304, 2018.
- [4] Q. Shi, S. Zhao, X. Cui, M. Lu, and M. Jia, "Anchor self-localization algorithm based on UWB ranging and inertial measurements," *Tsinghua Science and Technology*, vol. 24, no. 6, pp. 728–737, 2019.

- [5] A. D. Preter, J. Anthonis, and J. Swevers, "Experiment Design for Ultra-wideband Sensor Node Calibration," no. September, pp. 24–27, 2018.
- [6] W. Yu, X. Zhao, and G. Sun, "Self-calibration of anchor positions for indoor localization," *2017 IEEE International Conference on Robotics and Biomimetics, ROBIO 2017*, vol. 2018-Janua, pp. 581–586, 2018.
- [7] C. M. Almansa, W. Shule, J. P. Queralt, and T. Westerlund, "Autocalibration of a mobile UWB localization system for Ad-Hoc multi-robot deployments in GNSS-denied environments," *arXiv*, 2020.
- [8] M. Ridolfi, E. De Poorter, J. Fontaine, B. Van Herbruggen, W. Joseph, and J. Hoebeke, "UWB anchor nodes self-calibration in NLOS conditions : a machine learning and adaptive PHY error correction approach," *Wireless Networks*, vol. 0, 2021. [Online]. Available: <https://doi.org/10.1007/s11276-021-02631-0>
- [9] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.
- [10] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," *Proceedings - IEEE INFOCOM*, vol. 4, pp. 2652–2661, 2004.
- [11] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004*, 2004, pp. 46–54.
- [12] A. A. Kannan, G. Mao, and B. Vucetic, "Simulated annealing based localization in wireless sensor network," *Proceedings - Conference on Local Computer Networks, LCN*, vol. 2005, no. March 2017, pp. 513–514, 2005.
- [13] C. Savarese, J. Rabaey, and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," *Proceedings of the 2002 USENIX Annual Technical Conference*, no. May, 2002.
- [14] S. Čapkun, M. Hamdi, and J. P. Hubaux, "GPS-free positioning in mobile ad-hoc networks," *Proceedings of the Hawaii International Conference on System Sciences*, vol. 00, no. c, p. 255, 2001.
- [15] A. Ihler, J. Fisher, R. Moses, and A. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 809–819, 2005.
- [16] K. Whitehouse, C. Karlof, A. Woo, F. Jiang, and D. Culler, "The Effects of Ranging Noise on Multihop Localization."
- [17] A. Franzin and T. Stützle, "Revisiting simulated annealing: A component-based analysis," *Computers Operations Research*, vol. 104, 12 2018.
- [18] A. Kannan, B. Fidan, and G. Mao, "Analysis of flip ambiguities for robust sensor network localization," *Vehicular Technology, IEEE Transactions on*, vol. 59, pp. 2057 – 2070, 06 2010.
- [19] B. Van Herbruggen, B. Jooris, J. Rossey, M. Ridolfi, N. Macoir, Q. Van den Brande, S. Lemey, and E. De Poorter, "Wi-pos: A low-cost, open source ultra-wideband (uwb) hardware platform with long range sub-ghz backbone," *Sensors*, vol. 19, no. 7, 2019. [Online]. Available: <https://www.mdpi.com/1424-8220/19/7/1548>
- [20] S. L. Goh, G. Kendall, and N. Sabar, "Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem," *Journal of the Operational Research Society*, vol. 70, pp. 1–16, 06 2018.

Inhoudsopgave

Lijst van figuren	16
Lijst van tabellen	17
Lijst van algoritmes	18
Lijst van afkortingen	19
1 Inleiding	21
1.1 Inleiding tot indoorlokalisatie	21
1.2 Probleemstelling en doel van de masterproef	22
1.3 Structuur van de masterproef	22
2 Gerelateerd werk	23
2.1 Zelfkalibratie	23
2.1.1 Overzicht bestaande zelfkalibratie-algoritmes	23
2.1.2 Tekortkomingen	25
2.2 Multi-hop lokalisatie	25
2.2.1 Classificatie van multi-hop lokalisatie-algoritmes	26
2.2.2 Overzicht van multi-hop lokalisatie-algoritmes	27
2.3 Problemen bij lokalisatie	30

2.3.1	Afstandsbepalingsfouten	30
2.3.2	Positiebepalingsfouten	31
3	Simulated Annealing	34
3.1	Werking van het algoritme	35
3.2	Basiscomponenten van simulated annealing	36
3.2.1	Algoritme specifieke componenten	36
3.2.2	Probleemafhankelijke componenten	38
3.3	Motieven en afwegingen	40
3.3.1	Foutophoping	41
3.3.2	Globaal optimum	41
3.3.3	Anchorselectie	41
3.3.4	Schaalbaarheid en complexiteit	42
4	Overzicht testomgeving en werkwijze	43
4.1	Testomgeving	43
4.1.1	Overzicht testomgeving	43
4.1.2	Afstandsmetingen	44
4.1.3	Plaatsing van de anchors	46
4.1.4	Metrieken voor de plaatsing van de anchors	46
4.2	Werkwijze	48
5	Resultaten	50
5.1	Evaluatie simulated annealing algoritme	50
5.1.1	Flip ambiguïties	51

<i>INHOUDSOPGAVE</i>	15
5.1.2 Kostfunctie	53
5.1.3 Parameterinitialisatie	54
5.1.4 Anchorselectie	55
6 Conclusie	57
Bibliografie	59
Appendices	64

Lijst van figuren

2.1	Multipath propagation	30
2.2	Flip ambiguity	32
2.3	Flex ambiguity	32
4.1	Experiment setup with 15 anchors spread over an area of 40 x 25 m^2	44
4.2	Distribution of the absolute ranging error of the UWB measurements	45
4.3	Dilution of Precision - Heatmap	49
5.1	Correlation between localization error and number of known anchors	51
5.2	Estimated anchor locations	52
5.3	Estimated anchor locations - flip ambiguities	53
5.4	Correlation between cost and localization error	54
5.5	Localization Error	56

Lijst van tabellen

2.1	Overzicht zelfkalibratiemethodes	26
2.2	Overzicht van multi-hop localization algorithms	27
4.1	Overzicht van de anchorconfiguraties	46

Lijst van algoritmes

1	Simulated annealing localization - pseudocode formulation	35
2	Acceptance criteria - pseudocode formulation	39

Lijst van afkortingen

APS	Ad hoc positioning system
CIR	Channel impulse response
CF	Cost function
CM	Centroid method
DOP	Dilution of precision
DV	Distance-vector
GDOP	Geometric dilution of precision
GPS	Global positioning system
IMU	Inertial measurement unit
IPS	Indoor positioning system
LOS	Line-of-sight
LSE	Least squares estimator
MAC	Medium access control
MAE	Mean absolute error
MDS	Multidimensional scaling
MHLA	Multi-hop localization algorithm
MLE	Maximum likelihood estimation
MOCAP	Motion capture
NLOS	Non-line-of-sight
NUC	Next Unit of Computing
PRF	Pulse repetition frequency
RANSAC	Random sampling paradigm

RF	Radio frequency
SA	Simulated annealing
SDP	Semidefinite programming
SDS-TWR	symmetrical double-sided two-way ranging
SLAM	Simultaneous localization and mapping
TOA	Time of arrival
TX	Transmit
UWB	Ultra-wideband
WSN	Wireless sensor network

1

Inleiding

1.1 Inleiding tot indoorlokalisatie

Het doel van indoorlokalisatie is om de locatie van objecten of personen te bepalen op plaatsen waar het alombekende GPS en andere lokatiebepalingstechnieken onvoldoende nauwkeurigheid bieden. Indoorlokalisatie kan dan gebruikt worden om bijvoorbeeld passagiers te helpen in luchthavens, bezoekers rond te leiden in een museum of drones te voorzien van nauwkeurige locaties. Positiebepalingssystemen die werken op basis van satellieten zijn vaak niet bruikbaar in gebouwen omdat de satellietsignalen maar moeilijk door muren en plafonds kunnen dringen. Er bestaan verschillende andere technieken zoals Wi-Fi en Bluetooth die wel bruikbaar zijn in gebouwen en tot op enkele meters nauwkeurig zijn. Daarnaast bestaan er ook technieken op basis van ultra-wideband (UWB) die mogelijks nog beter geschikt zijn in dit type scenario's. UWB is gebaseerd op het versturen van een reeks extreem korte pulsen ($< 1\text{ns}$) die gespreid zijn over een grote bandbreedte. Deze korte pulsen maken het mogelijk om accuraat het tijdstip te bepalen waarop het bericht werd ontvangen. Dit maakt het mogelijk om afstanden tot op enkele centimeters nauwkeurig te bepalen. Vervolgens kunnen deze afstandsmetingen gebruikt worden om de positie te bepalen.

1.2 Probleemstelling en doel van de masterproef

Lokalisatiesystemen, al dan niet gebaseerd op UWB, maken doorgaans gebruik van anchors die zich op een vaste locatie bevinden. Eens de locaties van de anchors bekend zijn, kan het systeem tags lokaliseren die al dan niet in beweging zijn t.o.v. het referentiekader van de anchors. Het is van cruciaal belang dat de positie van deze anchors zo nauwkeurig mogelijk bepaald wordt, aangezien fouten in de positie van de anchors leiden tot een onnauwkeurige lokatiebepaling van de tag. Bij bestaande systemen wordt daarom vaak dure meetinfrastructuur gebruikt bij het installeren van de anchors. Het proces waarbij de locaties van de anchors bepaald worden is echter tijdrovend en foutgevoelig. Bovendien is het niet altijd mogelijk om deze kalibratie manueel te doen. De anchors kunnen zich bijvoorbeeld op moeilijk bereikbare plaatsen bevinden of er kunnen obstakels zijn die manuele opmetingen onmogelijk maken. Daarnaast zorgt deze manuele kalibratie voor extra kosten, zowel op vlak van meetinfrastructuur als op vlak van arbeidskrachten.

Om deze kosten te beperken en de gebruiksvriendelijkheid te verhogen, is het noodzakelijk om een zelfkalibratie-algoritme te ontwikkelen dat de locaties van de anchors automatisch kan bepalen. Deze algoritmes gebruiken hiervoor doorgaans de beschikbare afstandsmetingen die al dan niet werden bekomen gebruikmakend van UWB. De bestaande algoritmes zijn vaak slechts geëvalueerd aan de hand van simulaties of in kleinschalige *line-of-sight* (LOS) omgevingen. Het doel van deze masterproef is om een algoritme te ontwikkelen dat op grote schaal, waarbij niet elke anchor een directe link heeft met alle andere anchors, de locaties van de anchors kan bepalen. Het algoritme zal geëvalueerd worden in een real-life scenario in uitdagende NLOS-omstandigheden. Daarnaast wordt er onder meer onderzocht wat de invloed is van de positie en het aantal vooraf bekende anchors op de performantie.

1.3 Structuur van de masterproef

Het vervolg van deze masterproef wordt als volgt ingedeeld. In het volgende Hoofdstuk wordt een overzicht gegeven van relevant werk en de eventuele gebreken. Daarna wordt in Hoofdstuk 3 het algoritme in detail besproken. In Hoofdstuk 4 wordt een overzicht gegeven van de testomgeving. Daarnaast wordt ook de werkwijze toegelicht die gebruikt wordt voor de evaluatie van het algoritme. De resultaten worden in Hoofdstuk 5 geanalyseerd. Ten slotte worden in Hoofdstuk 6 de besluiten geformuleerd.

2

Gerelateerd werk

2.1 Zelfkalibratie

Positiebepalingstechnieken die werken op basis van UWB-afstandsmetingen moeten de positie van de anchors weten voordat ze de positie van de tags kunnen bepalen. Dit kan echter zeer resource- en arbeidsintensief zijn, zeker wanneer de ruimte groter wordt en er meer anchors nodig zijn. Doorgaans zijn hiervoor extra hulpmiddelen vereist zoals een laser- of landmeter.

Het doel van zelfkalibratie is om automatisch de positie van de anchors te bepalen. Deze procedure moet het mogelijk maken om het netwerk snel en dynamisch op te bouwen, zodat de kosten beperkt blijven en de gebruiksvriendelijkheid van het *indoor positioning system* (IPS) verhoogd wordt. Deze automatische kalibratie heeft ook een grote meerwaarde voor toepassingen die slechts tijdelijk nodig zijn of in situaties waar er onvoldoende tijd is, zoals bij reddingsoperaties. Daarnaast verkleint zelfkalibratie ook de kans op manuele fouten.

2.1.1 Overzicht bestaande zelfkalibratie-algoritmes

In deze sectie wordt een overzicht gegeven van de bestaande zelfkalibratietechnieken die zijn toegepast op positiebepaling. Dit overzicht is gebaseerd op de survey van Ridolfi et al. [1].

Bastone et al. [2] gebruiken een combinatie van het *random sampling paradigm* (RANSAC) en optimalisatietechnieken om een oplossing te vinden voor het TOA-zelfkalibratieprobleem. Eerst wordt er een robuuste methode gebruikt om een oplossing te vinden voor een kleine dataset. Vervolgens wordt deze oplossing sequentieel samengevoegd met de vorige oplossing. Deze oplossingsmethode werd getest in 2 testomgevingen waarbij de quadcopter die is uitgerust met een UWB-chip telkens verplaatst werd. De ene testomgeving was een MOCAP-studio van $2 \times 3 \times 2 \text{ m}^3$ waarbij de gemiddelde fout van de anchorposities ongeveer 12 cm bedraagt. De andere testomgeving was een grote ruimte van ongeveer $20 \times 20 \text{ m}^2$. De gemiddelde fout van de anchorposities voor deze opstelling bedraagt ongeveer 15 cm.

Hamer et al. [3] beschrijven een oplossingsmethode die ervoor zorgt dat de anchors hun eigen positie kunnen bepalen en deze vervolgens kunnen gebruiken om meerdere robots gelijktijdig te lokaliseren. Er wordt gestart met het definiëren van een coördinatensysteem. Daarbij wordt anchor A aangeduid als de oorsprong en anchors B, C en D moeten elk langs respectievelijk de positieve x-, y- en z-as liggen. De overige anchors mogen willekeurig geplaatst worden. Na het bepalen van het coördinatensysteem wordt de positie van de overige anchors bepaald met behulp van het gedistribueerde *gradient descent* algoritme. Het algoritme werd geëvalueerd in een testomgeving van $6 \times 7 \times 3.5 \text{ m}^3$ en had voor een opstelling van 8 anchors een RMSE (root-mean-square error) van 9.7 cm. Het nadeel van deze methode is dat de nauwkeurigheid van het systeem sterk afhankelijk is van de nauwkeurigheid van de afstandsmetingen tussen de 4 anchors die gebruikt worden om het coördinatensysteem te definiëren.

De zelfkalibratiemethode die door Shi et al. [4] wordt voorgesteld maakt naast UWB-metingen ook gebruik van een *inertial measurement unit* (IMU). De locaties van de anchors worden automatisch bepaald door een tag die zowel een UWB-chip als een IMU bevat vrij te verplaatsen in de testomgeving. Deze methode is gebaseerd op de SLAM-techniek waarbij de locatie van de anchors en de tag gelijktijdig kan worden bepaald. Het algoritme werd geëvalueerd door middel van simulaties. In de simulatieomgeving van $20 \times 20 \times 20 \text{ m}^3$ bedraagt de RMSE van de 4 anchors ongeveer 5 cm.

De Preter et al. [5] stellen ook een methode voor waarbij de kalibratie gebaseerd is op het verplaatsen van een tag. Deze keer is de tag niet uitgerust met een IMU, maar moet de tag een vooraf gedefinieerd pad afleggen. De testomgeving bestaat uit 4 anchors die zich op de hoekpunten van een ruimte van $8 \times 7 \text{ m}^2$ bevinden. Na een ruwe schatting van de anchorlocaties wordt het *maximum likelihood estimation* (MLE) algoritme gebruikt om deze schatting te verfijnen. In de experimenten werd er voor de 4 anchors een gemiddelde nauwkeurigheid behaald van minder dan 10 cm.

Yu et al. [6] beschrijven een zelfkalibratiemethode die bestaat uit 2 fasen. In de eerste fase wordt een iteratieve *centroid method* (CM) gebruikt om een initiële schatting te maken van de anchorlocaties. Vervolgens wordt in de 2e fase trilateration gebruikt om deze initiële schatting

te verbeteren. Deze methode werd getest in een ruimte van $12 \times 8 \text{ m}^2$ en de gemiddelde fout voor de 4 anchors bedraagt ongeveer 10 cm.

In tegenstelling tot de voorgaande methodes, stellen Almanase et al. [7] een kalibratieprocedure voor waarbij de anchors zich mogen verplaatsen. Bij het definiëren van het coördinatensysteem wordt een gelijkaardige techniek gebruikt als in [3]. Daarna wordt de positie van alle anchors aangepast door het verschil tussen de gemeten en de geschatte afstand te minimaliseren met een *least squares estimator* (LSE). Deze methode werd geëvalueerd in een testomgeving van 144 m^2 waarbij de gemiddelde fout voor de 4 anchors ongeveer 135 cm bedraagt.

De methodes die tot nu toe besproken werden, maken allemaal gebruik van UWB-metingen om de positie te bepalen. Geen van deze methodes maakt echter gebruik van de unieke UWB-eigenschappen. Ridolfi et al. [8] stellen een oplossing voor die naast de afstandsmetingen ook de unieke features van UWB gebruikt om zo de afstandsbepaling te verbeteren. Het voorgestelde framework maakt hiervoor onder meer gebruik van de *channel impulse responses* (CIRs). De verbeterde afstandsbepalingen worden vervolgens gebruikt door het *gradient descent* algoritme om de positie van de anchors te bepalen. Deze methode werd geëvalueerd in een testomgeving met een oppervlakte van $10 \times 4 \text{ m}^2$ en de gemiddelde fout voor de 4 anchors bedraagt 2.4 cm.

2.1.2 Tekortkomingen

Tabel 2.1 geeft een overzicht van de verschillende zelfkalibratiemethodes die eerder besproken werden. Deze oplossingen werden doorgaans geëvalueerd in LOS-omgevingen. Daarnaast was de grootte van de testomgeving eerder beperkt en werd er een laag aantal anchors gebruikt. In een kleinschalige omgeving worden meestal single-hop technieken gebruikt. Hierbij moet een anchor met een onbekende positie zich in het bereik van minstens 4 gelokaliseerde anchors bevinden voordat zijn positie kan worden bepaald. De bedoeling van deze thesis is om een eerste stap te zetten in de richting van zelfkalibratie-algoritmen voor grootschalige systemen. Bij een grotere oppervlakte verkleint echter de kans dat een anchor met een nog onbekende positie zich in het bereik van een gelokaliseerde anchor bevindt. Daarom wordt er in het volgende deel een overzicht gegeven van bestaande multi-hop algoritmes.

2.2 Multi-hop lokalisatie

Het doel van lokalisatie in wireless sensor networks (WSNs) is om de positie van nodes te bepalen gebruikmakend van beacons of anchors. Dit zijn meestal nodes waarvan de positie op voorhand bekend is. Het is gewenst om dit aantal te beperken, omdat een verlaging van het aantal beacons ook de kost van het positiebepalingssysteem doet dalen. Dit heeft echter als gevolg dat gewone

Tabel 2.1: Overzicht zelfkalibratiemethodes

Solution	Type	Condition	Algorithm	Anchors	Hop	Size	Accuracy
Bastone et al. [2]	Experimental	LOS	RANSAC	6	Single	$20 \times 20 \text{ m}^2$	15 cm
Hamer et al. [3]	Experimental	LOS	GD	8	Single	$6 \times 7 \times 3.5 \text{ m}^3$	9.7 cm
Shi et al. [4]	Simulation	LOS	SLAM	5	Single	$20 \times 20 \times 20 \text{ m}^3$	15 cm
De Preter et al. [5]	Experimental	LOS	MLE	4	Single	$8 \times 7 \text{ m}^2$	10 cm
Yu et al. [6]	Experimental	LOS	CM, trilateration	4	Single	$12 \times 8 \text{ m}^2$	10 cm
Almanasa et al. [7]	Simulation	LOS	LSE	4	Single	144 m^2	135 cm
Ridolfi et al. [8]	Experimental	LOS/NLOS	GD, UWB optimizations	4	Single	$10 \times 4 \text{ m}^2$	2.4 cm
Our solution	Experimental	LOS/NLOS	SA, UWB optimizations	15	Multi	$40 \times 25 \times 2.6 \text{ m}^3$	50 cm

nodes buiten het bereik van de anchors kunnen vallen waardoor ze nooit gelokaliseerd kunnen worden.

Om dit probleem op te lossen zijn er reeds verschillende multi-hop lokalisatie-algoritmes (MHLAs) ontwikkeld. Deze algoritmes vertonen fundamentele verschillen met de single-hop lokalisatie-algoritmes [9]. Bij single-hop algoritmes wordt aangenomen dat alle nodes in het netwerk direct kunnen communiceren met een verzameling van nodes waarvan de positie gekend is. Het doel van multi-hop algoritmes is om deze beperking weg te werken. Deze algoritmes introduceren daarnaast wel nieuwe uitdagingen en moeilijkheden die niet altijd gewenst zijn. Om ervoor te zorgen dat de nodes kunnen samenwerken, moet het bijvoorbeeld mogelijk zijn om informatie over grote afstanden te kunnen uitwisselen. Daarboven is het moeilijk om evaluaties op een kleine schaal te veralgemenen naar een grotere schaal. Bij single-hop algoritmes is dit wel mogelijk omdat elke groep van nodes als onafhankelijk kan worden gezien.

Deze uitdagingen zorgen ervoor dat het onderzoek naar MHLAs vooral gefocust is op theoretische analyse en simulaties. In dit deel worden eerst de verschillende criteria besproken die gebruikt worden bij de classificatie van MHLAs. Daarna wordt er een overzicht gegeven van bestaande MHLAs die de basis vormen voor andere algoritmes uit deze categorie. In het volgende hoofdstuk zal er vervolgens in meer detail worden gegaan op het geselecteerde algoritme.

2.2.1 Classificatie van multi-hop lokalisatie-algoritmes

Classificatie op basis van afstandsbepaling

Multi-hop lokalisatie-algoritmes (MHLAs) kunnen worden onderverdeeld op basis van de techniek die gebruikt wordt om de afstand te bepalen [10]. Een eerste categorie bestaat uit range-free methodes. Hierbij wordt de locatie van een node bepaald met behulp van niet-deterministische attributen zoals informatie over de nabijheid [11]. Deze algoritmes werken als volgt: een node die in het zendbereik van een andere node ligt, definieert een nabijheidsbeperking tussen beide nodes. Deze beperking kan dan verder gebruikt worden voor lokalisatie van andere nodes [12].

Een tweede categorie bestaat uit range-based methodes die in tegenstelling tot de range-free methodes wel beschikken over afstandsmetingen. Verschillende onderzoeken [13][14] hebben aangetoond dat range-based methodes in het algemeen nauwkeuriger zijn dan range-free methodes. Door hardwarebeperkingen is het echter niet altijd mogelijk om afstandsmetingen uit te voeren. In deze gevallen kunnen de range-free methodes toch gebruikt worden om op een kostefficiënte manier nodes te lokaliseren.

Classificatie op basis van de organisatie van de berekening

Een andere methode om MHLAs onder te verdelen is op basis van de organisatie van de berekening. Netwerken kunnen georganiseerd worden op een gecentraliseerde of een gedistribueerde manier. Gecentraliseerde netwerken verzamelen al de informatie in één centraal punt. Bij gedistribueerde netwerken worden de posities van de nodes daarentegen bepaald op basis van informatie die ze ontvangen van hun burens. De keuze voor een bepaalde categorie is meestal afhankelijk van de bestaande architectuur van de netwerken. Doorgaans is gedistribueerde lokalisatie moeilijker dan gecentraliseerde lokalisatie. Daarnaast kunnen gedistribueerde methodes steeds toegepast worden op gecentraliseerde problemen, terwijl dit omgekeerd niet het geval is [12].

2.2.2 Overzicht van multi-hop lokalisatie-algoritmes

In de volgende delen worden verschillende MHLA's besproken die als basis dienen voor andere MHLA's. In tabel 2.2 wordt er een overzicht gegeven van deze algoritmes. Dit overzicht is gebaseerd op het werk van Mao et al. [12].

Tabel 2.2: Overzicht van multi-hop localization algorithms

Solution	Topology	Range finding	Algorithm	Type
Bulusu et al. [15]	Distributed	Range-free	connectivity based	experimental
Niculescu et al. [16]	Distributed	Range-free	Ad Hoc Positioning System (APS)	simulation
Shang et al. [17]	Centralized	Range-free	multidimensional scaling (MDS)	simulation
Doherty et al. [18]	Centralized	Range-free	semi-definite programming (SDP)	simulation
Ji et al. [19]	Centralized	Range-based	multidimensional scaling (MDS)	simulation
Biswas et al. [20]	Centralized	Range-based	semi-definite programming (SDP)	simulation
Kannan et al. [21]	Centralized	Range-based	simulated annealing (SA)	simulation
Savarese et al. [22]	Distributed	Range-based	DV-distance	simulation
Capkun et al. [23]	Distributed	Range-based	local to global maps	simulation
Ihler et al. [24]	Distributed	Range-based	particle filters	simulation

Range-free - gedistribueerd

Bulusu et al. [15] stelden een range-free lokalisatiemethode voor die speciaal ontworpen is voor kleine apparaten met een laag energieverbruik. In plaats van extra hardware aan de bestaande sensoren toe te voegen, worden de bestaande RF-mogelijkheden gebruikt voor de lokalisatie. Het netwerk bestaat uit een vast aantal referentiepunten die periodiek beacon signalen verzenden. Om te vermijden dat de signalen van naburige referentiepunten overlappen, wordt er aangenomen dat de referentiepunten gesynchroniseerd zijn. De nodes die gelokaliseerd moeten worden verzamelen de ontvangen signalen van de referentiepunten en lokaliseren zichzelf in de regio waarin het zendbereik van de referentiepunten elkaar overlappen.

In [16] introduceerden Niculescu et al. een gedistribueerd lokalisatie-algoritme voor dat als een uitbreiding werkt op distance vector routing en gps-lokalisatie. In de eerste fase van het algoritme worden de locaties van alle anchors verspreid over alle andere nodes in het netwerk. Elk bericht bevat naast de locaties van de anchors ook een hop-teller. Wanneer een anchor een bericht van een andere anchor ontvangt, dan wordt de gemiddelde hop-afstand geschat en verspreid over het netwerk als een correctiefactor. Zodra een niet-anchor node 3 correctiefactoren ontvangen heeft, kan deze triangulatie gebruiken om zijn positie te bepalen.

Range-free - gecentraliseerd

Shang et al. [17] ontwikkelden een oplossing die gebaseerd is op multidimensional scaling (MDS). MDS is een techniek die gebruikt wordt bij data-analyse om informatie over de nabijheid om te zetten in een abstract cartesisch coördinatenstelsel. Eerst worden de kortste paden berekend die dan gebruikt kunnen worden om de afstandsmatrix op te stellen. Op deze matrix wordt daarna MDS toegepast, zodat een benadering van de relatieve coördinaten van de nodes kan worden bekomen. In de laatste stap worden de relatieve coördinaten omgezet naar absolute coördinaten door te zorgen dat de benaderde coördinaten samenvallen met de absolute coördinaten. De benaderde locaties kunnen verfijnd worden door least-squares minimization te gebruiken [12].

In [25] hebben Shang et al. hun algoritme verbeterd met behulp van de MDS-MAP lokalisatiemethode. Het netwerk wordt onderverdeeld in verschillende overlappende regio's, zodat het MDS-algoritme eerst lokaal kan worden uitgevoerd. Later worden de resultaten van elke regio opnieuw samengevoegd in 1 globale map. Simulaties tonen aan dat de performantie in vergelijking met de gecentraliseerde methode behouden blijft en dat het zelfs betere resultaten geeft bij netwerken met een onregelmatige vorm.

Range-based - gedistribueerd

In de volgende delen worden er enkele range-based lokaliserings-algoritmes besproken. Deze zijn vaak een uitbreiding op de range-free algoritmes die in het vorige deel aan bod kwamen. Het DV-hop algoritme van Niculescu et al. [16] werd door dezelfde onderzoekers uitgebreid met afstandsmetingen. Dit DV-distance algoritme heeft een gelijkaardige werking als het DV-hop algoritme, met als verschil dat nu de gemeten afstand over het netwerk verspreid wordt in plaats van het aantal hops. In [22] werd een algoritme voorgesteld dat bestaat uit 2 fasen. De bedoeling van de eerste fase is om een schatting te bekomen van alle onbekende locaties. In de tweede fase worden de afstandsmetingen tussen de burens gebruikt om de locaties uit de vorige fase te verfijnen met behulp van least-squares trilateration. Naast het verfijnen van de locatie wordt er voor elke node ook een extra waarde toegevoegd die aangeeft hoe betrouwbaar de geschatte locatie is. Deze waarde varieert van 0 tot 1, waarbij 1 staat voor een betrouwbare schatting.

In [19] en [23] wordt een methode gebruikt die al eerder aan bod kwam bij MDS-MAP. Deze methodes construeren eerst verschillende lokale mappen van nodes, waarna ze worden samengevoegd tot 1 globale map. De grootste problemen bij deze methodes zijn de verspreiding van de fouten en het grote aantal iteraties dat vereist is om ervoor te zorgen dat het algoritme convergeert [12].

Range-based - gecentraliseerd

De gecentraliseerde distance-based lokalisatie-algoritmes kunnen worden onderverdeeld in 3 categorieën: multidimensional scaling (MDS), stochastic optimization en linear programming. Ook hier zijn deze algoritmes vaak een uitbreiding op de range-free algoritmes die eerder zijn besproken. Zo werd de MDS-methode uit [17] uitgebreid met afstandsmetingen door Ji et al. [19]. Het netwerk wordt opnieuw onderverdeeld in verschillende overlappende regio's. Elke regio bevat minimaal 3 nodes waarvan de locatie reeds bepaald is. Daarna wordt MDS gebruikt binnen elke regio gebruikt om de relatieve coördinaten van de nodes te bekomen en na het samenvoegen ontstaat er opnieuw 1 globale map.

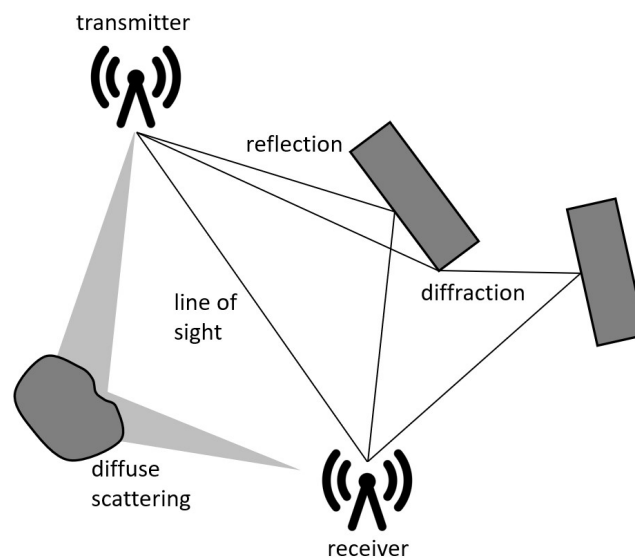
Kannan et al. [21] stelden een methode voor die gebaseerd is op simulated annealing. De naam van dit optimalisatiealgoritme komt van het annealing proces uit de metaalkunde: hierbij wordt een metaal opgewarmd en vervolgens gecontroleerd afgekoeld om zo het metaal sterker te maken. Er wordt een kostfunctie gedefinieerd die aangeeft hoe goed de geschatte posities zijn. Het doel van deze methode is om op zoek te gaan naar het globaal minimum, zonder vast te blijven zitten in een lokaal minimum. In de beginfase heeft het systeem een hoge temperatuur, omdat de coördinaten van de onbekende nodes willekeurig bepaald worden. In elke stap van het algo-

ritme wordt er een niet-anchor node verplaatst over een steeds kleiner wordende afstand in een willekeurige richting. Daarna wordt opnieuw de totale kost berekend. Wanneer deze kleiner is dan de kost vóór de verplaatsing, dan wordt deze verandering altijd aanvaard. In het andere geval kan het zijn dat deze ook nog aanvaard wordt, maar die kans verkleint steeds naarmate de temperatuur van het systeem afneemt. Deze methode zal in Hoofdstuk 3 in detail worden besproken.

2.3 Problemen bij lokalisatie

2.3.1 Afstandsbepalingsfouten

Ultra-wideband systemen zijn zeer populair bij indoorlokalisatiesystemen dankzij de unieke signaaleigenschappen. UWB is gebaseerd op het versturen van een reeks extreem korte pulsen ($< 1\text{ns}$) die gespreid zijn over een grote bandbreedte. Deze korte pulsen maken het mogelijk om accuraat het tijdstip te bepalen waarop het bericht werd ontvangen. Hierdoor kan de afstand tot op enkele centimeters nauwkeurig bepaald worden. Er zijn echter ook verschillende andere factoren die de kwaliteit van de afstandsmetingen negatief beïnvloeden. Aangezien de prestatie van de positiebepalingsalgoritmen sterk afhankelijk is van deze afstandsmetingen, is het belangrijk om te begrijpen door welke factoren deze metingen beïnvloed worden. MacGougan et al. [26] beschrijven verschillende mogelijke foutbronnen. Sidorenko et al. [27] bestuderen verschillende methodes om deze fouten te corrigeren. In Figuur 2.1 wordt een overzicht gegeven van verschillende fouten die kunnen optreden tijdens de propagatie van het draadloze signaal van de zender naar de ontvanger.



Figuur 2.1: Multipath propagation

In het ideale scenario zijn er geen obstakels en komt het verzonden signaal rechtstreeks aan bij de ontvanger. In werkelijkheid zorgt de interactie van het signaal met zijn omgeving ervoor dat het signaal meerdere keren wordt ontvangen. De verzameling van signalen die elk een verschillend pad volgen van de zender naar de ontvanger worden *multipath* componenten genoemd [28]. Verschillende mogelijk interacties van een signaal met zijn omgeving zijn:

- Geen interactie (*line-of-sight*): in LOS-omstandigheden waarbij er zich geen obstakels bevinden op het directe pad tussen de zender en de ontvanger, kan het signaal zich zonder hinder voortplanten
- Reflectie (*reflection*): bij objecten met een glad oppervlak zoals muren en vloeren zal het signaal symmetrisch gereflecteerd worden
- Verstrooiing (*diffuse scattering*): bij interactie van het signaal met een objecten met een onregelmatig oppervlak zal het signaal in verschillende richtingen verstrooid worden
- Doordringing (*penetration*): afhankelijk van het materiaal waaruit het obstakel bestaat, zal naast het gereflecteerde signaal ook een deel van het signaal doordringen in het object

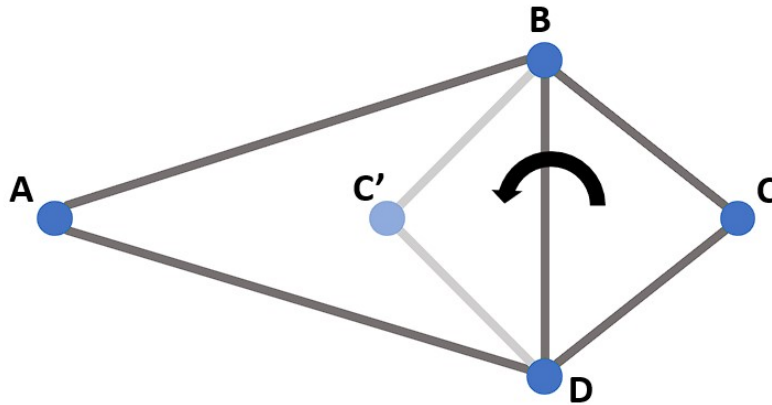
Saunders et al. [29] gaan dieper in op de verschillende foutbronnen bij RF-gebaseerde afstandsbepalingstechnieken. In het vervolg van deze masterproef ligt de nadruk eerder bij de positiebepaling, maar het is dus belangrijk om te onthouden dat de performantie van positiebepalingsalgoritmen sterk afhankelijk is van de kwaliteit van de afstandsmetingen.

2.3.2 Positiebepalingsfouten

Nauwkeurige afstandsmetingen bieden nog geen garantie op een nauwkeurige positiebepaling. Naast meetfouten kunnen er nog andere fundamentele problemen optreden waardoor er geen unieke oplossing bestaat voor een bepaald sensornetwerk [12]. Er bestaan verschillende frameworks uit de grafentheorie die kunnen worden gebruikt om na te gaan of er voor een gegeven netwerk een unieke oplossing bestaat. Daarbij wordt er vooral gekeken naar hoe rigide een graaf is met behulp van de afstandsinformatie. Een graaf is globaal rigide als en slechts als 2 mogelijke grafen voor dezelfde verzameling van afstanden slechts verschillen door een combinatie van translatie, rotatie en reflectie. Er kunnen 2 type *ambiguities* (dubbelzinnigheden) gedefinieerd worden:

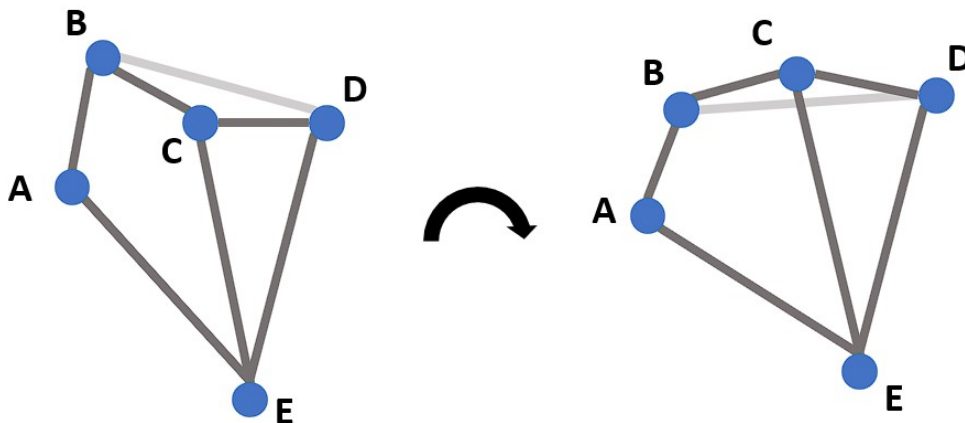
- flip ambiguity
- flex ambiguity

In Figuur 2.2 wordt een voorbeeld van flip ambiguity weergegeven. De blauwe punten stellen nodes voor en de grijze verbindingslijnen geven aan dat de afstand tussen de 2 nodes gekend is. Wanneer dezelfde verzameling van afstanden gebruikt wordt, is het mogelijk om een graaf te vormen waarbij node C gespiegeld wordt over verbindingslijn (B,D). Het is voor deze graaf niet mogelijk om de positie van node C eenduidig te bepalen.



Figuur 2.2: Flip ambiguity

Figuur 2.3 geeft een graaf weer waarbij een ander type ambiguity kan optreden: flex ambiguity. Stel dat je de verbindingslijn (B,D) tijdelijk wegneemt, dan kunnen nodes A, B en C verplaatst worden (met behoud van de onderlinge afstanden tussen de nodes) waardoor de structuur van de graaf zodanig verandert dat de oorspronkelijke graaf niet meer kan worden bekomen door een combinatie van translatie, rotatie of reflectie. Aangezien het na deze verplaatsing nog steeds mogelijk is om de verbindingslijn (B,D) terug te plaatsen met dezelfde lengte is er dus te weinig informatie gekend om de positie van de nodes eenduidig te bepalen.



Figuur 2.3: Flex ambiguity

Kannan et al. [30] geven een overzicht van relevant onderzoek over ambiguïties. De nadruk van deze studies ligt voornamelijk bij flip ambiguïties, aangezien dit type probleem doorgaans vaker voorkomt in vergelijking met flex ambiguïties [30] en vaak gepaard gaat met grote fouten bij de positiebepaling. Bestaande oplossingen voor flip ambiguïties maken gebruik van de unit disk graph [31] om het bereik van de draadloze nodes voor te stellen. Bij dit model heeft elke node één cirkel met een vaste straal die afhankelijk is van het zendbereik van de node. Er wordt aangenomen dat 2 nodes verbonden zijn wanneer ze zich in elkaars bereik bevinden. Het probleem is dat het zendbereik niet duidelijk kan worden bepaald. Dit kan veroorzaakt worden door omgevingsfactoren zoals gebouwen of interne factoren zoals de beschikbaarheid van energie.

Flip ambiguïties komen meestal voor wanneer de burens van een node zich ongeveer op 1 lijn bevinden of wanneer de afstandsmetingen onnauwkeurig zijn. Daarnaast kunnen deze fouten zich snel verspreiden in de volgende iteraties van positiebepalingsalgoritmen waardoor grote delen van het netwerk negatief kunnen worden beïnvloed. Een empirische studie van Kannan et al. toont aan dat de kans op flip ambiguïties toeneemt naarmate het aantal burens afneemt. Dit kan een grote impact hebben in de beginfase van de algoritmes wanneer er nog niet veel locaties gekend zijn.

3

Simulated Annealing

Simulated Annealing (SA) is een stochastisch algoritme dat vanuit een initiële oplossing zijn omgeving gaat verkennen om zo een benadering van het globale optimum van een gegeven functie te vinden. Vanuit de huidige oplossing wordt een nieuwe oplossing geconstrueerd. Hierbij wordt een kostfunctie gebruikt als indicatie voor de kwaliteit van de oplossing. Doorgaans wordt een betere oplossing (met een lagere kost) steeds geaccepteerd. Er bestaat ook een kans dat een mindere oplossing geaccepteerd wordt, maar deze kans neemt af naarmate de temperatuur afneemt.

Dit optimalisatiealgoritme is gebaseerd op een techniek uit de metaalbewerking: uitgloeien. Hierbij wordt een metaal verhit en daarna gecontroleerd afgekoeld met als doel het aantal kristallen te vergroten en zo het metaal sterker te maken. Bij hoge temperaturen kunnen de moleculen vrij bewegen en treden er grote verandering op aan de structuur van het metaal. Deze temperatuur neemt in de loop van de tijd af samen met de kans dat een molecule zal bewegen.

Eerst zal de basiswerking van het SA-algoritme besproken worden. Vervolgens wordt er dieper ingegaan op de werking van het algoritme met behulp van een op componenten gebaseerde aanpak. Ten slotte wordt verduidelijkt waarom dit algoritme gekozen werd en welke gevolgen deze keuze heeft.

3.1 Werking van het algoritme

De basiswerking van het algoritme wordt in pseudocode weergegeven in Algoritme 1. Dit is slechts een mogelijke opbouw aangezien er voor verschillende stappen een alternatieve implementatie bestaat.

Algorithm 1: Simulated annealing localization - pseudocode formulation

Input: distance measurements, coordinates of known anchors, room dimensions

Output: estimated position for each anchor

```

1 Create initial solution  $S$ 
2 Set temperature  $T$ 
3 Set move distance  $d$ 
4 while StopConditions not met do
5   for each anchor with an unknown position do
6     // move the anchor of solution  $S^*$  by a distance  $d$  in an arbitrary direction
7     Create new solution  $S'$ 
8     if AcceptanceConditions met then
9       // depending on the difference in cost and the temperature  $T$ 
10      Accept  $S'$ 
11    else
12      Reject  $S'$ 
13  Update  $T$  // according to the cooling schedule
14  Update  $d$ 
15 Print solution  $S^*$ 

```

Het algoritme start met het creëren van een initiële oplossing (*lijn 1*). Dit is een oplossing die aan alle geldigheidsvoorwaarden voldoet en relatief eenvoudig te construeren is, maar zelden de beste oplossing zal zijn. Vervolgens wordt de temperatuur (*lijn 2*) en de verplaatsingsafstand (*lijn 3*) ingesteld. Vanuit deze initiële oplossing wordt er gezocht naar een nieuwe kandidaatoplossing (*lijn 6*) door de buurt van de oplossing te verkennen op basis van de verkenningscriteria. Zodra er een nieuwe oplossing gevonden wordt, dan wordt deze geëvalueerd met behulp van de aanvaardingscriteria (*lijn 7*). Doorgaans worden betere oplossingen steeds aanvaard. Er bestaat ook een kans dat een minder goede oplossing aanvaard wordt, maar die kans verkleint naarmate de temperatuur van het systeem afneemt. Deze temperatuur verandert volgens een vooraf gedefinieerd koelschema (*lijn 11*). Na elke iteratie van het algoritme worden de stopvoorwaarden (*lijn 4*) geëvalueerd. Indien aan deze voorwaarden voldaan zijn, stopt het algoritme en wordt de beste kandidaatoplossing teruggegeven als resultaat (*lijn 13*).

3.2 Basiscomponenten van simulated annealing

Het bovenstaande algoritme kan worden onderverdeeld in verschillende componenten. Elke component kan gezien worden als een parameter waardoor er nieuwe varianten van het algoritme kunnen worden gecreëerd. Franzin et al. [32] stellen de volgende 9 componenten voor die samen een SA-algoritme definiëren:

1. Initiële oplossing
2. Kandidaatoplossing
3. Verkenningscriteria
4. Aanvaardingscriteria
5. Begintemperatuur
6. Temperatuurlengte
7. Koelschema
8. Opnieuw verhitten
9. Stopvoorwaarden

De eerste 2 componenten zijn probleemafhankelijk, terwijl de laatste 7 componenten eigen zijn aan het algoritme. In dit deel worden eerst de 7 algoritme specifieke componenten besproken. Daarna volgt een beschrijving van de 2 probleemafhankelijke componenten die noodzakelijk zijn om het positiebepalingsprobleem op te lossen.

3.2.1 Algoritme specifieke componenten

Verkenningscriteria

Deze component moet op zoek gaan naar nieuwe kandidaatoplossingen. Traditionele SA-implementaties doen dit door een willekeurige verkenning van de omgeving uit te voeren. Deze methode wordt door de grote meerderheid van SA-implementaties gebruikt [32]. Het nadeel van deze techniek is dat potentiële verbeteringen kunnen worden mislopen. Dit komt omdat de kans dat een oplossing aanvaard wordt afneemt naarmate de temperatuur afneemt en dat het tegelijkertijd moeilijk kan zijn om te ontsnappen uit een lokaal optima [33].

Om deze problemen te vermijden werden er verschillende alternatieve oplossingen [33][34] ontwikkeld. De techniek die in deze implementatie gebruikt wordt is gebaseerd op de verbeteringen

die worden voorgesteld door Ishibuchi et al. [34]. Om ervoor te zorgen dat het algoritme zo veel mogelijke de omgeving kan verkennen, worden er k willekeurige oplossingen gegenereerd. Daarna wordt de beste oplossing vergeleken met de huidige oplossing. Een lichte variant van deze methode gaat tussenoplossingen die beter zijn dan de huidige oplossing meteen aanvaarden.

Aanvaardingscriteria

Deze component bepaalt of een kandidaatoplossing al dan niet aanvaard wordt. Hiervoor wordt vaak een kostfunctie gebruikt als indicatie voor de kwaliteit van de oplossing. Doorgaans duidt een lagere kost op een betere oplossing. De meerderheid van de aanvaardingscriteria volgen een eenvoudig patroon waarbij betere oplossingen steeds aanvaard worden. Minder goede oplossingen worden slechts sporadisch aanvaard op basis van specifieke criteria. Door soms toch een minder goede oplossing te aanvaarden, kan het algoritme de huidige regio kan verlaten en nieuwe regio's verkennen die mogelijk betere oplossingen opleveren.

Een van de meest voorkomende aanvaardingscriteria is gebaseerd op de Metropolis-voorwaarde [35] die werd voorgesteld bij de originele formulering van het SA-algoritme [36]. Daar wordt een betere oplossing steeds aanvaard. Een slechtere oplossing kan ook aanvaard worden, maar die kans verkleint naarmate het verschil in kost tussen de huidige en de nieuwe oplossing stijgt. In [32] geven Franzin et al. een overzicht van alternatieve aanvaardingscriteria.

De moeilijkheid bij deze criteria is dat het niet eenvoudig is om een goede balans te vinden tussen het aanvaarden en verwerpen van oplossingen. Wanneer je enerzijds te veel slechtere oplossingen verwerpt, blijft het algoritme mogelijk vastzitten in een lokaal optimum. Bij het aanvaarden van te veel slechtere oplossingen bestaat anderzijds het risico dat je de huidige oplossing onvoldoende verbetert. Deze balans wordt doorgaans bepaald door parameters waarvan er vaak geen duidelijke richtlijnen zijn over hoe deze moeten worden ingesteld. Een automatische instelling/dynamische verandering van deze parameters kan mogelijk een oplossing bieden (zie Sectie 5.1.3).

Begintemperatuur

De aanvaardingscriteria uit de vorige component werken meestal op basis van de temperatuurparameter. Deze component zorgt ervoor dat de begintemperatuur wordt ingesteld. Dit kan zijn door reeds bekende informatie over het systeem te gebruiken, maar ook door bijvoorbeeld eerst een willekeurige verkenning van de omgeving uit te voeren.

Temperatuurlengte

Deze component bepaalt hoeveel nieuwe oplossingen er gezocht worden voordat de temperatuur wordt geüpdatet. Een eerste mogelijkheid is om de temperatuur te updaten nadat een vast aantal oplossingen onderzocht werden. Een andere mogelijkheid is om dit aantal te wijzigen op basis van de grootte van de buurt.

Koelschema

Het koelschema geeft aan hoe de temperatuur zal evolueren naarmate het algoritme vordert. In de standaardimplementatie neemt de temperatuur monotoon af waardoor de kans dat een minder goede oplossing geaccepteerd wordt ook steeds afneemt.

Opnieuw verhitten

In de meeste implementaties neemt de temperatuur geleidelijk af. Dit kan ervoor zorgen dat er na een bepaalde tijd nauwelijks nog slechtere oplossingen aanvaard worden. Daarom bestaat er ook de mogelijkheid om de temperatuur te resetten naar een hogere waarde zodra de temperatuur een minimale waarde bereikt heeft.

Stopvoorwaarden

De stopvoorwaarden bepalen wanneer het algoritme beëindigd wordt. Dit kan gebeuren na een bepaalde tijd, of na een vast aantal iteraties. Om te vermijden dat de uitvoering van het algoritme vroegtijdig wordt afgebroken op basis van vooraf bepaalde voorwaarden, is het ook mogelijk om deze voorwaarden dynamisch aan te passen. Het algoritme kan bijvoorbeeld ook stoppen wanneer er na een bepaald aantal iteraties geen verbeteringen gevonden zijn of wanneer het percentage van betere oplossingen onder een gegeven drempelwaarde vallen.

3.2.2 Probleemafhankelijke componenten

In dit deel wordt er besproken hoe simulated annealing kan worden toegepast op het lokaliseringsprobleem. Deze methode is gebaseerd op het schema dat door Kannan et al. [21] wordt voorgesteld. Zoals eerder aangegeven kan het algoritme worden onderverdeeld in 9 componenten. De eerste 7 componenten werden reeds besproken. Nu zal er meer in detail worden gegaan op de laatste 2 probleemspecifieke componenten, namelijk het genereren van een initiële oplossing en het zoeken naar kandidaatoplossingen.

Initiële oplossing

Deze component wordt gebruikt in de beginfase van het algoritme en zorgt ervoor dat alle nodes, behalve de nodes waarvan de locatie vooraf gekend is, geïnitieerd worden met een willekeurige locatie die binnen de opgegeven grenzen van het systeem (de afmetingen van de ruimte) liggen. Het algoritme vereist dat de locatie van minimaal 3 nodes vooraf gekend is.

Kandidaatoplossing

Er wordt een nieuwe kandidaatoplossing gecreëerd door een anchor (waarvan de locatie nog niet op voorhand gekend is) in een willekeurige richting te verplaatsen over een steeds kleiner wordende afstand. Deze verplaatsingsafstand kan net zoals de temperatuur een afstandsverminderingsschema hebben die ervoor zorgt dat de afstand na elke iteratie wordt geüpdatet. Uiteindelijk bepalen de opgestelde aanvaardingscriteria of een kandidaatoplossing aanvaard wordt. De werking van het aanvaardingschema wordt in pseudocode weergegeven in Algoritme 2.

Algorithm 2: Acceptance criteria - pseudocode formulation

Input: temperature T , CF_{new} , CF_{old}

Output: True/False

```

1  $\Delta(CF) = CF_{new} - CF_{old}$ 
2 if  $\Delta(CF) \leq 0$  then
   | // Downhill move  $\Rightarrow$  accept always
3 | Return True
4 else if random number from uniform distribution  $(0,1) \leq e^{-\Delta(CF)/T}$  then
   | // Uphill move  $\Rightarrow$  accept with probability
5 | Return True
6 else
   | // Reject candidate
7 | Return False

```

Dit schema gebruikt een kostfunctie als indicator voor de kwaliteit van de oplossing. De vergelijking van de kostfunctie (CF) wordt weergegeven in Vergelijking 3.1.

$$CF = \sum_{i=1}^N \sum_{j \in N_i} (d'_{ij} - d_{ij})^2 \quad (3.1)$$

$$d'_{ij} = \sqrt{(x'_i - x'_{ij})^2 + (y'_i - y'_{ij})^2 + (z'_i - z'_{ij})^2}$$

N : totaal aantal onbekende anchors

N_i : verzameling van de buren van anchor i

d'_{ij} : geschatte afstand tussen anchor i en buur j

d_{ij} : gemeten afstand tussen anchor i en buur j

(x'_i, y'_i, z'_i) : geschatte coördinaten van anchor i

$(x'_{ij}, y'_{ij}, z'_{ij})$: geschatte coördinaten van buur j van anchor i

Anchor's zijn buren wanneer ze zich in elkaars zendbereik bevinden. Merk op dat de verzameling N_i dus zowel anchors kan bevatten waarvan de locatie vooraf bekend is, als anchors waarvan de locatie nog moet worden bepaald. Voor elke nieuwe kandidaatoplossing wordt de kost opnieuw berekend. Deze oplossing zal steeds aanvaard worden wanneer het verschil in kost tussen de nieuwe en de oude oplossing kleiner of gelijk is aan nul (Vergelijking 3.2).

$$\Delta(CF) = CF_{new} - CF_{old} \quad (3.2)$$

Indien $\Delta(CF) \geq 0$, dan bepaalt de Metropolis-voorwaarde of deze minder minder goede oplossing toch aanvaard wordt. Dit is dus afhankelijk van de temperatuur, het bijhorende afkoelingsschema en $\Delta(CF)$. Doorgaans worden de temperatuur en het afkoelingsschema empirisch bepaald. In Sectie 5.1.3 wordt er dieper ingegaan op deze parameterinitialisatie.

3.3 Motieven en afwegingen

De 3 hoofdreden waarom dit algoritme gekozen is, zijn:

- de weerstand tegen foutophoping als gevolg van fouten bij de afstands- en positiebepaling;
- de mogelijkheid om te ontsnappen uit een lokaal optimum waardoor het globaal optimum mogelijk beter kan worden benaderd;
- de vrijheid bij de anchorselectie.

Daarnaast zijn er ook enkele nadelen waarvan de invloed afhankelijk is van de toepassing:

- de schaalbaarheid en complexiteit;
- de parameterinitialisatie.

3.3.1 Foutophoping

Bestaande zelfkalibratietechnieken (zie Sectie 2.1.1) maken doorgaans gebruik van single-hop algoritmen omdat ze slechts getest werden in een kleine ruimte met een beperkt aantal anchors. Naarmate het positiebepalingssysteem een grotere oppervlakte moet dekken, verkleint de kans dat een anchor met een nog onbekende positie zich in het bereik van een gelokaliseerde anchor bevindt. Om dit probleem op te lossen zijn er reeds verschillende multi-hop lokalisatie-algoritmes ontwikkeld (zie Sectie 2.2.2). De belangrijkste beperking bij multihop-methoden is dat er een ongunstige trend kan optreden waarbij fouten bij de afstands- en positiebepaling worden opgestapeld [37]. Een kleine fout aan het begin kan enkele hops verder een veel grotere fout veroorzaken. Kannan et al. [21] hebben aangetoond dat deze fouten voortplanting niet of nauwelijks optreedt bij het simulated annealing positiebepalingsalgoritme.

3.3.2 Globaal optimum

Naarst de weerstand tegen foutophoping is het algoritme ook goed bestand tegen het convergeren naar een vals lokaal optimum [12]. Andere oplossingsmethoden zoals het gradient descent algoritme veranderen de oplossing enkel in de richting waar de kost lager is. Bij elke wijziging beweegt het algoritme bergafwaarts. Dit kan ertoe leiden dat de oplossing vastloopt in een lokaal minimum. Simulated annealing staat toe dat oplossingen zich op een gecontroleerde manier ook bergopwaarts kunnen bewegen. Hierdoor is het mogelijk om uit een lokaal minimum te ontsnappen en zo een potentieel betere oplossing te vinden.

3.3.3 Anchorselectie

Dit multi-hop algoritme geeft ook meer vrijheid bij de selectie en plaatsing van de anchors. De bestaande zelfkalibratietechnieken stellen meestal een referentiesysteem op door eerst de oorsprong A van het systeem te definiëren. Vervolgens bepalen de posities van anchors B en C respectievelijk de positieve x- en y-as. Daarnaast wordt er ook aangenomen dat de z-as loodrecht staat op het vlak dat door de punten A, B en C gaat. De posities van de anchors kunnen dan als volgt geïnitieerd worden [3]:

$$(x_A, y_A, z_A) = (0, 0, 0)$$

$$(x_B, y_B, z_B) = (d_{AB}, 0, 0)$$

$$(x_C, y_C, z_C) = \left(\frac{x_B^2 + d_{AC}^2 - d_{BC}^2}{2x_B}, \sqrt{d_{AC}^2 - x_C^2}, 0 \right)$$

Deze techniek is echter sterk afhankelijk van de nauwkeurigheid van de afstandsmetingen tussen anchors A, B en C. Kleine fouten in deze afstandsmetingen zorgen ervoor dat het referentiesysteem niet correct gedefinieerd kan worden. Dit kan leiden tot grote fouten bij de positiebepaling van de overige nodes in het netwerk.

Deze problemen kunnen worden vermeden met het zelfkalibratie-algoritme dat gebaseerd is op simulated annealing door de positie van minstens 3 anchors op voorhand in te geven. Dit is vooral zinvol wanneer de omgeving semi-gestructureerd is. Zo delen verschillende anchors vaak dezelfde x-, y- of z-coördinaat omdat ze werden vastgemaakt aan dezelfde muur of plafond. In sommige situaties is er zelfs al een referentiesysteem gedefinieerd. Het volstaat dan om de positie van slechts 3 anchors te bepalen in het bestaande referentiesysteem. Vervolgens kan het algoritme automatisch de positie van de overige anchors bepalen zonder dat er daarna coördinatentransformaties nodig zijn.

3.3.4 Schaalbaarheid en complexiteit

Tijdens de evaluatie van het algoritme in Hoofdstuk 5 zal de nadruk voornamelijk liggen bij de nauwkeurigheid van de positiebepaling. Daarnaast zijn er nog verschillende andere elementen waarvan de relevantie meestal afhankelijk is van de toepassing. Voorbeelden hiervan zijn [38]:

- schaalbaarheid;
- rekencomplexiteit;
- overhead bij de communicatie;
- kost van de hardware.

Het simulated annealing algoritme heeft een gecentraliseerd schema waardoor het doorgaans een nauwkeurigere positiebepaling heeft in vergelijking met gedistribueerde methodes. Dit zorgt er echter voor dat de schaalbaarheid van het algoritme eerder beperkt is [12]. Bovendien neemt de rekencomplexiteit van gecentraliseerde algoritmes vaak sneller toe naarmate de grootte van het netwerk toeneemt. Het is soms ook gewenst om de communicatie tussen de anchors te beperken, aangezien het zenden en ontvangen van berichten meestal de grootste energieverbruikers zijn [38].

4

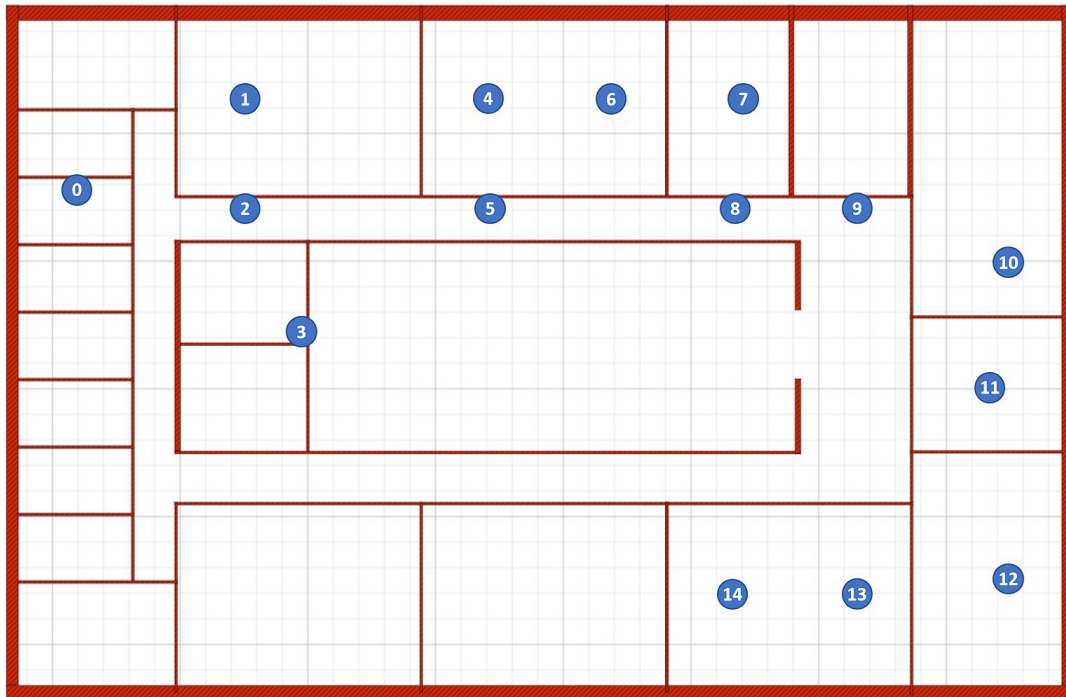
Overzicht testomgeving en werkwijze

In dit hoofdstuk wordt eerst een overzicht gegeven van de testomgeving. Vervolgens wordt de werkwijze besproken die gebruikt wordt voor de evaluatie van het algoritme.

4.1 Testomgeving

4.1.1 Overzicht testomgeving

De testomgeving bestaat uit 15 anchors die verspreid zijn over een oppervlakte van ongeveer $40 \times 25 \text{ m}^2$. Deze experimentele opstelling bevindt zich in een kantooromgeving waarbij muren en andere obstakels zorgen voor uitdagende NLOS-omstandigheden. Het aantal burens van een anchor varieert van 3 tot 12. Geen enkele anchor kan dus alle andere anchors bereiken. Een anchor heeft gemiddeld 7 burens waarvan er meestal slechts 1 buur is die zich in het directe gezichtsveld (LOS) van de anchor bevindt. In Figuur 4.1 wordt de layout van de anchors in de testomgeving weergegeven. De anchors zijn vastgemaakt aan het plafond in verschillende kantoren, vergaderruimtes en gangen op verdieping 9 van het iGent gebouw. Elke anchor bestaat uit een Intel NUC die is uitgerust met een UWB-chip waarvan de hardware uitgebreid wordt besproken in [39].



Figuur 4.1: Experiment setup with 15 anchors spread over an area of $40 \times 25 \text{ m}^2$

4.1.2 Afstandsmetingen

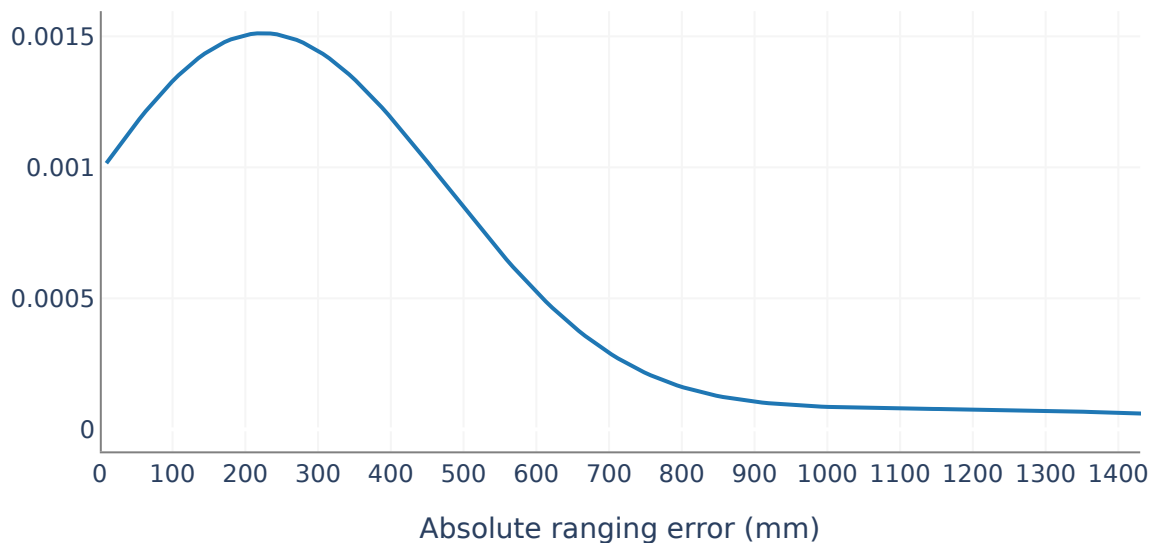
De anchors kunnen de afstand tot hun directe burens bepalen met behulp van ultra-wideband. Deze UWB-afstandsbepalingstechnologie is goed bestand tegen multipath-effecten en NLOS-omstandigheden [40]. UWB is gebaseerd op het versturen van een reeks extreem korte pulsen ($< 1\text{ns}$) die gespreid zijn over een grote bandbreedte. Hierdoor is het mogelijk om accuraat het tijdstip te bepalen waarop het signaal ontvangen werd. De verzamelde tijdstippen kunnen met behulp van de SDS-TWR-techniek [41] (symmetrical double-sided two-way ranging) worden omgezet in een nauwkeurige afstandsbepaling wanneer wordt aangenomen dat de voortplantingsnelheid van het UWB-signaal bekend is. Om interferentie te vermijden zal aan elke anchor het kanaal voor een vaste tijdspanne toegekend worden. Het MAC-protocol dat in deze testomgeving gebruikt wordt om deze toegang te regelen, wordt in detail besproken in [42].

Deze techniek geeft goede resultaten in LOS-omstandigheden en is doorgaans tot op enkele centimeters nauwkeurig. De performantie kan echter snel afnemen in NLOS-omstandigheden, zeker wanneer de UWB-instellingen niet zijn aangepast aan de omgeving [8].

Elke anchor gebruikt de volgende UWB-instellingen:

- Kanaal: 3
- Bitrate: 110 kbps
- Preamble: 4096 symbolen
- PRF: 64 MHz
- TX gain: Default (11.5 dB)

Deze instelling werd gekozen omdat eerdere experimenten in deze testomgeving hebben aangetoond dat dit resulteerde in de meest betrouwbare afstandsmetingen. Figuur 4.2 geeft een overzicht van de absolute fout tussen de geschatte afstand (UWB) en de gemeten afstand (laser). Hierbij worden de afstanden die bekomen werden met behulp van een lasermeter gezien als de *ground truth*. Merk op dat deze lasermetingen die tot op enkele millimeters nauwkeurig zijn wel gevoeliger zijn voor manuele fouten. Hierdoor zijn ze minder betrouwbaar dan een ground truth die bijvoorbeeld werd bekomen met behulp van een MOCAP-systeem.



Figuur 4.2: Distribution of the absolute ranging error of the UWB measurements

Voor elke combinatie van 2 anchors die zich in elkaars bereik bevinden, werd de mediaan van ongeveer 500 afstandsmetingen gebruikt als input voor het algoritme. Daarnaast werden er ook verschillende testen uitgevoerd waarbij de afstandsmetingen gecorrigeerd werden met behulp van gelijkaardige machine learning modellen als deze die besproken worden in [43]. Deze correcties werden telkens toegepast op een verschillend aantal metingen: 1, 10 en 100. Afhankelijk van de anchorconfiguratie zorgde dit voor een kleine verbetering van het eindresultaat. Aangezien

de verschillen tussen deze 4 datasets eerder beperkt zijn, zal enkel de eerste dataset gebruikt worden voor de validatie van het algoritme. Hoewel er eerder een grote verbetering verwacht werd na het corrigeren van de afstandsmetingen, is deze beperkte verandering van het resultaat niet geheel onverwacht. Severi et al. [44] geven namelijk aan dat het verzamelen en verbeteren van de afstandsmetingen niet meer de belangrijkste fase is van MHLAs. Daarentegen moet er eerder gefocust worden op de positiebepaling [44].

4.1.3 Plaatsing van de anchors

Naast de nauwkeurigheid van de afstandsmetingen heeft de plaatsing van de anchors ook een significante invloed op de nauwkeurigheid van de positiebepaling [45][46]. Du et al. [47] hebben aangetoond dat het patroon waarbij de anchors gelijkzijdige driehoeken vormen de beste resultaten geeft in WSNs. Door obstakels zoals kasten en muren is het echter niet altijd mogelijk om anchors op de meest ideale posities te plaatsen. Daarom worden er voor de validatie van het algoritme in totaal 30 verschillende anchorconfiguraties getest. Er zijn 10 anchorconfiguraties (3A-3J) waarbij de positie van 3 anchors op voorhand gekend is. In configuraties 4A-4J en 5A-5J worden deze uitgebreid met respectievelijk 1 en 2 willekeurig geselecteerde anchors. De verschillende configuraties worden weergegeven in Tabel 4.1.

Tabel 4.1: Overzicht van de anchorconfiguraties

Configuration	Number of Anchors		
	3	4	5
A	0,1,2	0,1,2,4	0,1,2,4,5
B	4,5,6	4,5,6,8	4,5,6,8,7
C	6,8,9	6,8,9,10	6,7,8,9,10
D	9,10,11	9,10,11,12	8,9,10,11,12
E	11,12,14	11,12,13,14	10,11,12,13,14
F	0,9,11	0,4,9,11	0,4,9,11,14
G	2,6,9	2,6,9,12	2,6,9,12,14
H	1,6,10	1,6,10,13	1,5,6,10,13
I	4,6,7	1,4,6,7	1,4,6,7,8
J	5,8,9	2,5,8,9	0,2,5,8,9

4.1.4 Metrieken voor de plaatsing van de anchors

De verschillende anchorconfiguraties worden geëvalueerd aan de hand van de volgende 3 metrieken: anchorspreiding, connectiviteit en *geometric dilution of precision* (GDOP).

Anchorspreiding

Een eerste metriek om de anchorconfiguraties onder te verdelen in verschillende categorieën steunt op de spreiding van de anchors. Een veelvoorkomend voorstel is om de anchors zo ver mogelijk uit elkaar te plaatsen, indien mogelijk aan de randen van het netwerk [46]. Er wordt verwacht dat een grotere spreiding van de anchors zal resulteren in een kleinere positiebepalingfout gemiddeld gezien voor het volledige netwerk. Om deze metriek te berekenen wordt de som van de afstanden tussen de anchorparen als indicator gebruikt.

Connectiviteit

Bij deze tweede metriek gebeurt de categorisering op basis van de connectiviteit en de gemiddelde fout per anchor. De verwachting is dat het bij de anchorselectie vermoedelijk beter is om te kiezen voor anchors die veel burens hebben. Het idee hierachter is dat de foutenophoping en -verspreiding beperkt zal blijven omdat er zo een groter aantal anchors met een nog onbekende positie al minstens 1 anchor hebben waarvan de positie bekend is.

Een ander item dat vermoedelijk een nauw verband houdt met deze metriek is de gemiddelde fout per anchor en zijn single-hop burens. Indien er bij de planning van het netwerk bijvoorbeeld verwacht wordt dat de afstandsmetingen van een bepaalde anchor onnauwkeurig zijn omdat de omgeving van deze anchor voor veel interferentie zorgt, is het dan verstandig om de locatie van deze anchor manueel op te meten of is het beter om enkel vooraf gekende anchors in te geven die betrouwbare afstandsmetingen hebben? De verwachting is dat de connectiviteit minder van belang zal zijn in NLOS-omstandigheden omdat de afstandsmetingen in deze omgeving doorgaans minder nauwkeurig zijn. Zodra een anchor een minimumaantal burens heeft, wordt verwacht dat de nauwkeurigheid van die burens een grotere rol zal spelen dan het aantal burens.

Deze metriek wordt voorgesteld door het aantal unieke single-hop burens van elke anchor met een bekende locatie. Daarnaast wordt ook de som genomen van de gemiddelde fout bij de UWB-afstandsmetingen per anchor om te onderzoeken of er rekening moet worden gehouden met de betrouwbaarheid van deze metingen tijdens de anchorselectie.

Geometric Dilution of Precision

In tegenstelling tot de voorgaande metrieken zal deze laatste metriek wel rekening houden met de geometrie van de geselecteerde anchors. Hiervoor wordt de Geometric Dilution of Precision (GDOP) berekend. De term dilution of precision (DOP) is afkomstig van satellietnavigatiesystemen zoals GPS, Galileo en GLONASS [48]. In het ideale geval zal een kleine meetfout slechts

leiden tot slechts een kleine fout bij de positiebepaling. Bij een slechte geometrie van de anchors kan het systeem echter gevoeliger zijn voor meetfouten waardoor kleine meetfouten grote positiebepalingsfouten kunnen veroorzaken. Het doel van de GDOP-waarde is om op basis van de relatieve geometrie van satellieten (of anchors in het geval van UWB) aan te geven welke accuraatheid er mag verwacht worden bij de positiebepaling [49].

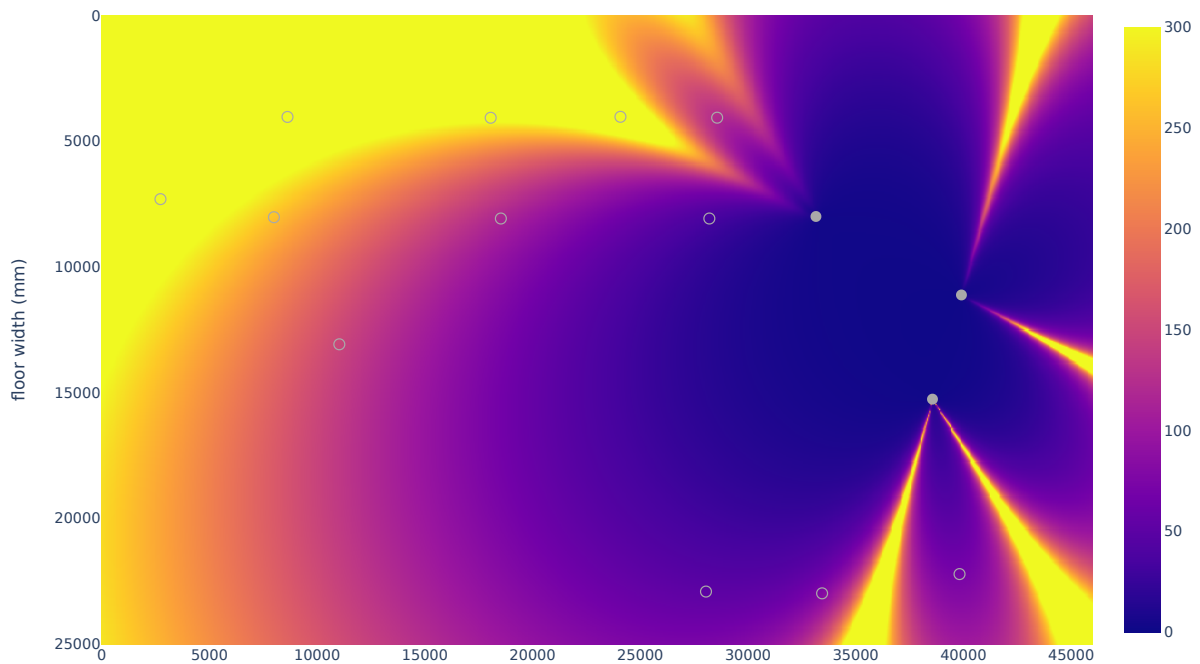
Dit principe wordt visueel voorgesteld door de heatmaps in Figuur 4.3. De locaties van de bekende anchors worden voorgesteld door een volle grijze cirkel, terwijl de overige anchors worden voorgesteld door een lege cirkel. Een lage GDOP-waarde (blauw) geeft aan dat de geometrie van de bekende anchors op dit punt zorgt voor een betrouwbare positiebepaling. Een hoge GDOP-waarde (geel) geeft aan dat deze anchorgeometrie zeer gevoelig is voor meetfouten en daardoor mogelijk een minder betrouwbare positiebepaling zal opleveren.

4.2 Werkwijze

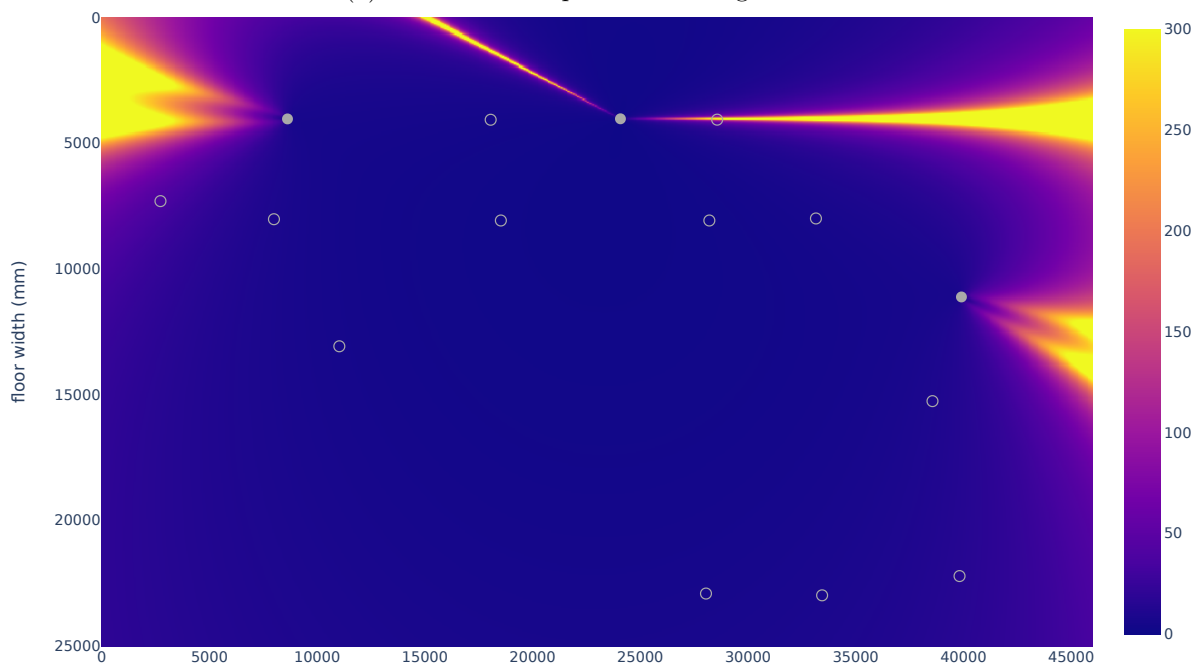
Nu de gehele testomgeving bekend is samen met de metrieken die gebruikt worden voor de evaluatie, zal dit deel ingaan op de werkwijze. De afstandsmetingen uit sectie 4.1.2 worden eerst centraal opgeslagen in een CSV-bestand. Elke rij bevat de afstandsinformatie van 1 paar nodes. Vervolgens wordt deze informatie ingelezen door het algoritme uit Hoofdstuk 3. Daarnaast worden ook de z -coördinaten van alle anchors ingegeven, maar dit is niet vereist. Deze keuze werd gemaakt omdat de z -coördinaten van de anchors nauwelijks verschillen van elkaar ($\Delta z = z_{max} - z_{min} < 10 \text{ cm}$).

Dit algoritme is geïmplementeerd in C++ en wordt uitgevoerd op een virtuele machine. Deze heeft een Intel® Xeon® Platinum 8168 processor met een basis kloksnelheid van 2.7 GHz en een maximale single-core frequentie van 3.7 GHz. De VM draait op Ubuntu 20.04 en beschikt naast 2 vCPU's ook over 4 GiB RAM. De uitvoering van 1 simulatie neemt na het inlezen van de afstandsmetingen uit de testomgeving 57s in beslag (*logging* niet inbegrepen). Deze tijd is sterk afhankelijk van de begintemperatuur en het koelschema, aangezien het algoritme pas stopt wanneer de opgegeven minimumtemperatuur bereikt is. Het is mogelijk om de uitvoering van het algoritme te versnellen, maar tijdens de experimenten ligt de focus eerder op het eindresultaat (de geschatte posities) dan op de uitvoeringstijd.

Na de uitvoering van het algoritme worden de geschatte posities vergeleken met de ground truth posities, zodat de gemiddelde absolute fout (MAE) berekend kan worden. De ground truth werd bekomen door afstandsmetingen uit te voeren met een lasermeter. Deze metingen zijn gevoelig voor manuele fouten, maar zorgen nog steeds voor milimeter nauwkeurigheid. De resultaten van deze simulaties worden in Hoofdstuk 5 besproken.



(a) GDOP heatmap - anchorconfiguration 3D



(b) GDOP heatmap - anchorconfiguration 3H

Figuur 4.3: (a) en (b) stellen de heatmaps voor van respectievelijk configuraties 3D en 3H. De heatmap van configuratie D bevat zichtbaar meer gele zones die overeenkomen met een hoge GDOP. Dit doet vermoeden dat configuratie H mogelijk beter is omdat de gele kleur daar minder dominant aanwezig is. In Hoofdstuk 5 zal worden besproken of dit ook werkelijk het geval is en wat de mogelijke gevolgen zijn.

5

Resultaten

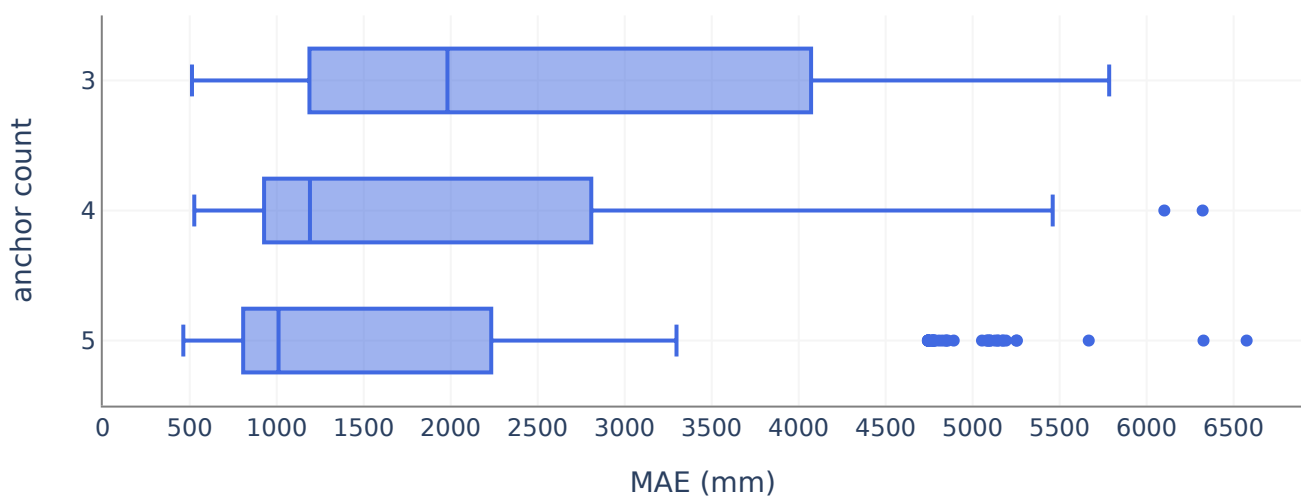
In dit hoofdstuk wordt het simulated annealing algoritme uit Hoofdstuk 3 geëvalueerd in de testomgeving die besproken werd in Hoofdstuk 4. In totaal werden 30 verschillende anchorconfiguraties getest, waarbij de opstelling van de anchors en de bijhorende afstandsmetingen steeds hetzelfde bleven. Het enige verschil tussen de configuraties is de selectie van de op voorhand bekende anchors. Eerst wordt onderzocht wat de invloed is van het aantal op voorhand bekende anchors op de gemiddelde absolute fout (MAE) bij de positiebepaling. Vervolgens wordt er meer in detail gegaan op de problemen die kunnen optreden tijdens de uitvoering van het zelfkalibratie-algoritme. Hierbij worden zowel problemen besproken die eigen zijn aan het algoritme (kostfunctie, parameterinitialisatie), als problemen die ook kunnen optreden bij andere positiebepalingsalgoritmes (flip ambiguities).

5.1 Evaluatie simulated annealing algoritme

Nadat de basiswerking van het algoritme geverifieerd werd aan de hand van de lasermetingen die gebruikt werden als ground truth, kon het algoritme toegepast worden op de UWB-afstandsmetingen. Zoals eerder aangegeven in Hoofdstuk 3 vereist het algoritme dat de locatie van minstens 3 anchors op voorhand gekend is. Dit aantal varieerde van 3 tot 5, afhankelijk van de anchorconfiguratie. Daarnaast verwacht het algoritme ook alle beschikbare afstandsmetingen

tussen de anchors en hun directe buren. Deze metingen vonden plaats in NLOS-omstandigheden waarbij muren en andere obstakels grote afwijkingen kunnen veroorzaken. Dit resulteerde in een gemiddelde fout van 337 mm.

Figuur 5.1 geeft de MAE weer waarbij de verschillende anchorconfiguraties gegroepeerd werden op het aantal op voorhand bekende anchors. Elke groep bestaat uit 10 anchorconfiguraties en per configuratie werden 20 simulaties uitgevoerd. Voor de configuraties met 3 bekende anchors bedraagt de MAE 1981 mm (mediaan). Deze fout neemt af tot 1010 mm naarmate het aantal bekende anchors toeneemt. Deze daling is minder opvallend aanwezig bij de vergelijking van de minima tussen de verschillende groepen.

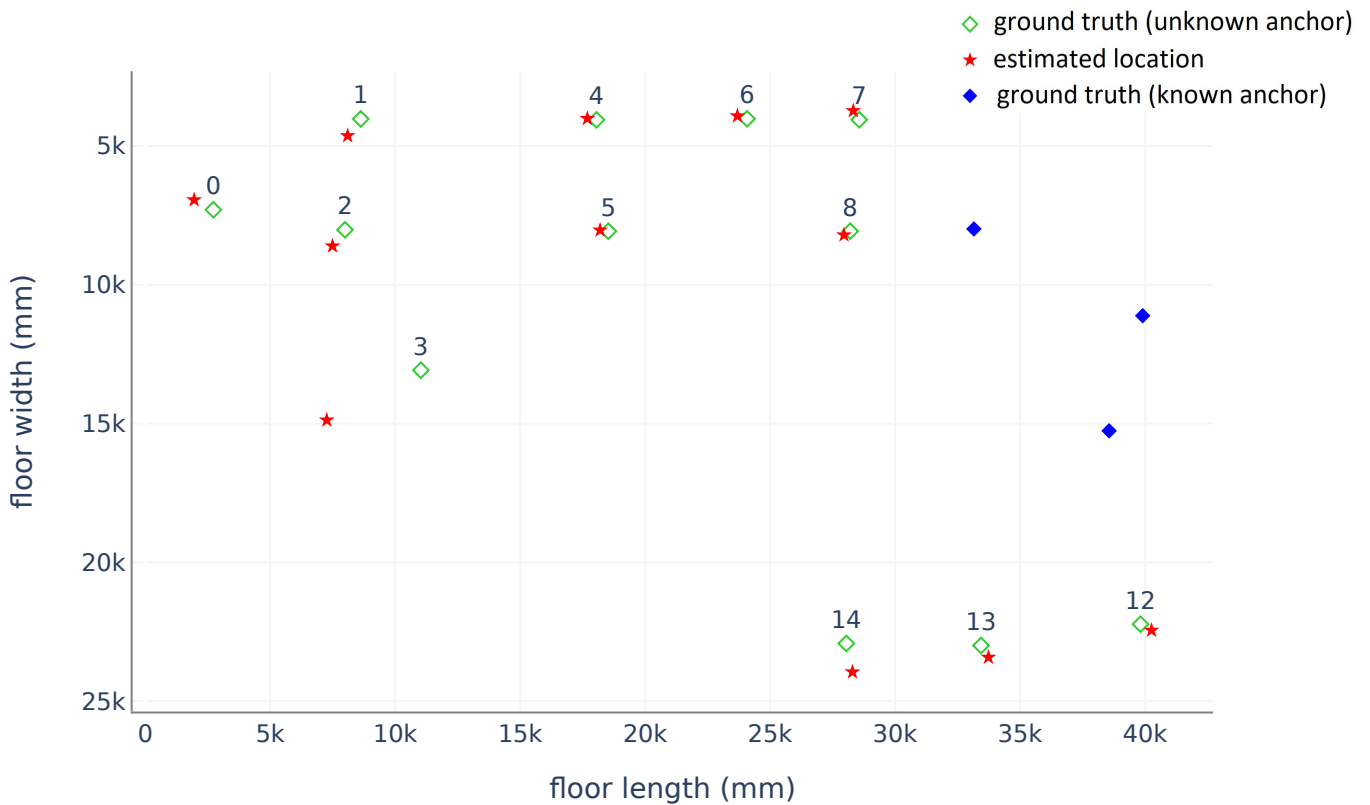


Figuur 5.1: Correlation between localization error and number of known anchors

De minima en medianen liggen in de lijn van de verwachtingen, maar de spreiding en de maxima zijn echter hoger dan verwacht. De specifieke problemen en mogelijke oplossingen worden in de volgende delen van dit hoofdstuk besproken.

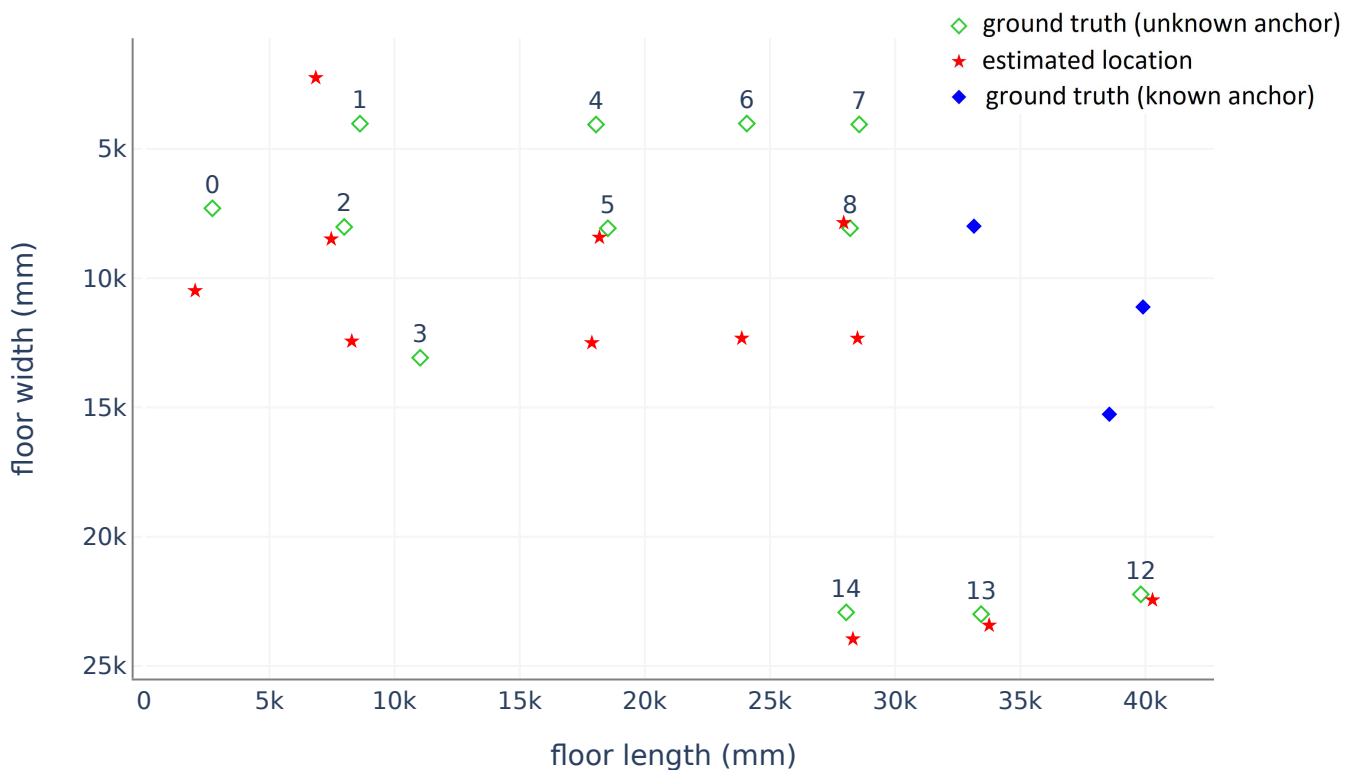
5.1.1 Flip ambiguities

De vermoedelijke hoofdoorzaak van de hoge spreiding en maxima is de aanwezigheid van flip ambiguities. Dit type probleem werd eerder al besproken in Sectie 2.3.2 en komt vaak voor wanneer de buren van een anchor zich nagenoeg op 1 lijn bevinden. In Figuur 5.2 wordt het bovenaanzicht van de testopstelling weergegeven waarbij de werkelijke en de geschatte locaties respectievelijk worden voorgesteld door een ruit en een ster.



Figuur 5.2: Estimated anchor locations

Voor deze specifieke simulatie bedraagt de MAE 870 mm. Dit resultaat ligt in lijn met de verwachtingen. Voor een andere simulatie van dezelfde anchorconfiguratie met exact dezelfde afstandsmetingen bedraagt de MAE echter 4 330 mm. De geschatte posities van deze simulatie worden weergegeven in Figuur 5.3. Het merkwaardige aan deze resultaten is dat de geschatte posities van anchors 4, 6 en 7 lijken te zijn gespiegeld om hun burens (2, 5 en 8). De oorzaak hiervan ligt hoogstwaarschijnlijk bij de geometrie van de anchors en de onzekerheid bij de positiebepaling van de burens. Zodra de positie van 1 van de burens gespiegeld is, kan dit een lawine-effect veroorzaken waarbij de burens van een gespiegelde anchor op hun buurt ook gespiegeld kunnen worden.



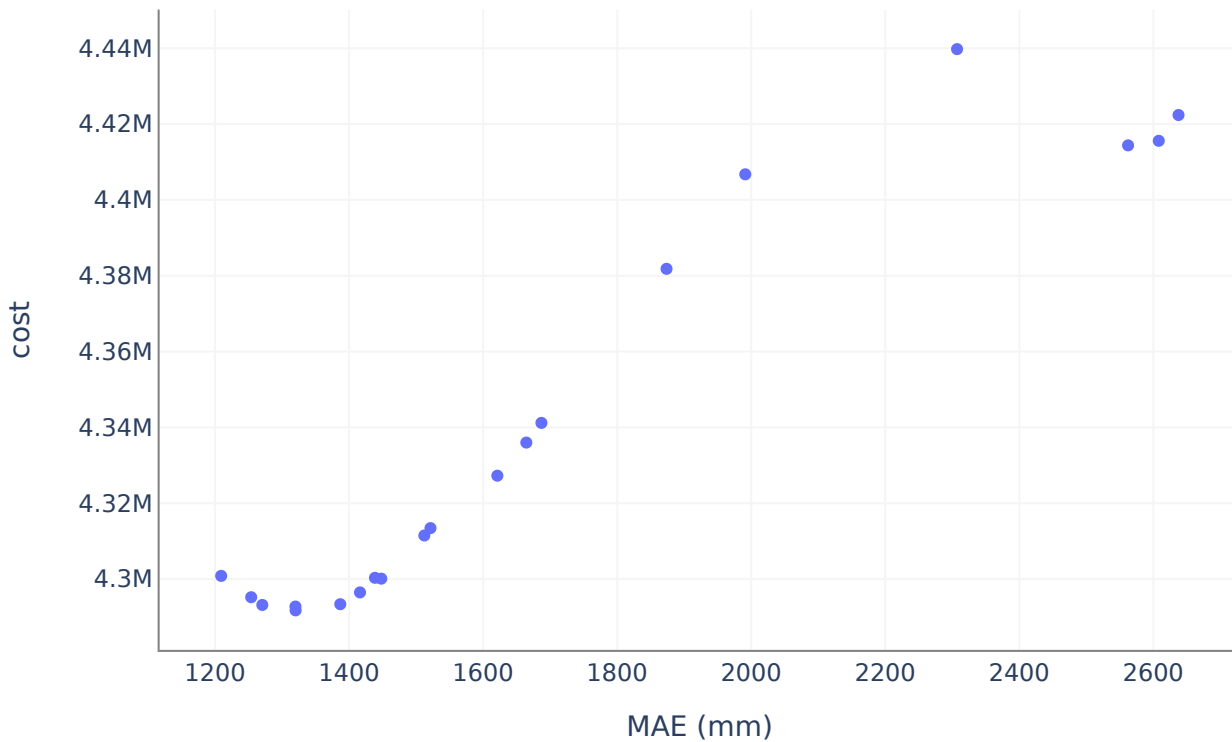
Figuur 5.3: Estimated anchor locations - flip ambiguities

5.1.2 Kostfunctie

Een andere factor die een grote invloed heeft op het eindresultaat is de kostfunctie. Deze functie wordt gebruikt als indicator voor de kwaliteit van een oplossing. Het doel van het simulated annealing algoritme is om het globaal minimum van deze functie te vinden. In Figuur 5.4 wordt het verband weergegeven tussen de kost en de MAE voor 20 simulaties.

Over het algemeen duidt een lagere kost op een betere oplossing wat de gemiddelde absolute fout betreft. Er zijn echter enkele simulaties die hiervan afwijken en ondanks een lagere kost toch een hogere absolute fout hebben.

Een eerste factor die ervoor kan zorgen dat de kostfunctie niet representatief is voor de kwaliteit van de oplossing is de aanwezigheid van flip ambiguïties. De kost wordt bepaald door het verschil te berekenen tussen de werkelijke en de geschatte afstand van een anchor met zijn burens. Stel dat een anchor 3 burens heeft die zich nagenoeg op 1 lijn bevinden, dan kan die anchor gespiegeld worden om deze lijn zonder dat dit een invloed heeft op de kost. Een mogelijke verbetering voor



Figuur 5.4: Correlation between cost and localization error

de kostfunctie is om naast het verschil in afstand ook rekening te houden met de geometrie van de buren.

Een andere factor die de effectiviteit van de kostfunctie beïnvloed is de nauwkeurigheid van de afstandsmetingen. Naast de logische oplossing om de afstandsmetingen te verbeteren kan de performantie mogelijk verbeterd worden door een betrouwbaarheidsfactor toe te kennen aan elke afstandsmeting. Daarbij kunnen bijvoorbeeld de buren waarvan de ontvangen signaalsterkte het hoogst is een hogere betrouwbaarheid krijgen.

5.1.3 Parameterinitialisatie

Naast het detecteren of vermijden van flip ambiguïties en het verbeteren van de kostfunctie zijn er ook verbeteringen mogelijk bij de initialisatie van de parameters. Een van de belangrijkste parameters is de temperatuur T . Deze parameter speelt samen met het koelschema een grote rol bij het aanvaarden en verwerpen van nieuwe oplossingen. In de literatuur bestaan er slechts enkele vuistregels voor het geschikt instellen van deze parameters [50]. Zo wordt er bijvoorbeeld

aangeraden om de initiële temperatuur in te stellen zodat de kans op het accepteren van een minder goede oplossing (met een hogere kost) ongeveer 80% bedraagt [21]. Door de willekeurige plaatsing van de anchors in de beginfase van het algoritme kunnen er echter grote verschillen ontstaan in de initiële kost van de opstelling waardoor het niet eenvoudig is om een geschikte begintemperatuur te vinden. Een mogelijke oplossing hiervoor is om de parameters automatisch in te stellen afhankelijk van de behaalde kost na enkele verkorte testsimulaties.

Het koelschema kan eventueel dynamisch worden aangepast afhankelijk van het aantal gevonden oplossingen met een lagere kost. Indien er in de afgelopen simulaties bijvoorbeeld steeds een betere oplossing gevonden wordt, is het misschien overbodig om ook mindere oplossingen te aanvaarden. Zo zal het algoritme sneller een (lokaal) minima bereiken. Vervolgens kan *reheating* [51] gebruikt worden om de temperatuur opnieuw te verhogen om op die manier te ontsnappen uit dit lokaal minima.

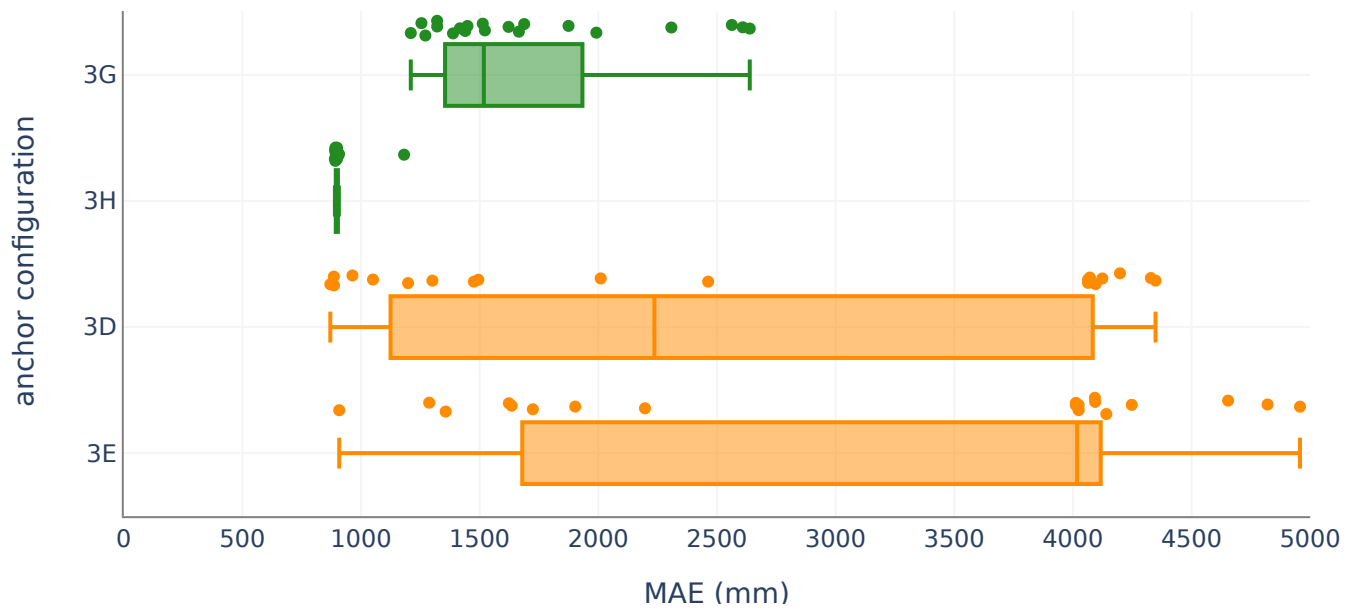
5.1.4 Anchorselectie

In totaal werden 30 verschillende anchorconfiguraties getest. Aan het begin van dit hoofdstuk werden deze configuraties gegroepeerd op het aantal vooraf bekende anchors. Ondanks het feit dat voor elke simulatie dezelfde afstandsmetingen gebruikt werden, bestaan er toch grote verschillen tussen de verschillende configuraties met hetzelfde aantal vooraf bekende anchors. In Figuur 5.5 wordt de MAE weergegeven voor 4 configuraties waarbij telkens de posities van 3 anchors op voorhand bekend zijn. 2 van deze configuraties hebben een relatief lage MAE (3G, 3H), de andere 2 configuraties hebben een relatief hoge MAE (3D, 3E). In Bijlage A wordt een overzicht gegeven van de MAE voor elke configuratie afzonderlijk.

Er is een duidelijk merkbaar verschil tussen de MAE voor deze configuraties. Dit toont aan dat niet alleen de plaatsing van de anchors belangrijk is, maar dat de keuze van de vooraf bekende anchors ook een grote invloed heeft op het resultaat. Om te onderzoeken waarom bepaalde configuraties betere resultaten opleveren, werden de volgende metrieken gebruikt:

- Anchorspreiding: geeft aan hoe ver de anchors zich van elkaar bevinden
- Anchorconnectiviteit: indicator voor de netwerkdensiteit in de buurt van een anchor
- GDOP: indicator voor de kwaliteit van de configuratie op basis van de relatieve geometrie van de gekende anchors

Deze metrieken werden eerder besproken in Hoofdstuk 4. Er is echter geen duidelijk verband gevonden tussen bovenstaande metrieken en de MAE. De GDOP-waarde lijkt de beste indicator



Figuur 5.5: Localization Error

te zijn om te bepalen hoe goed de geselecteerde anchors zijn, maar aangezien dit algoritme slechts werd getest in één unieke testomgeving is hiervoor verder onderzoek nodig.

6

Conclusie

Dit zelfkalibratie-algoritme dat gebaseerd is op simulated annealing is robuust en kan de posities van de anchors bepalen in een realistische testomgeving in NLOS-omstandigheden. Ondanks een gemiddelde afstandsbepalingsfout van 34 cm met uitschieters tot 452 cm, kan het algoritme de foutenophoping beperkt houden. Dit resulteert in een MAE die varieert van 50 cm tot 250 cm afhankelijk van de anchorconfiguratie.

Een van de oorzaken van een hoge MAE is de aanwezigheid van flip ambiguïties. Er bestaan reeds verschillende technieken die deze ambiguïties kunnen verminderen, maar deze zijn minder geschikt in NLOS-omstandigheden. Dit komt omdat de bestaande technieken meestal gebaseerd zijn op het zendbereik, maar dit kan sterk variëren door de aanwezigheid van obstakels. De aanwezigheid van flip ambiguïties kan verminderd worden door het algoritme meerdere keren uit te voeren en vervolgens enkel verder te gaan met het beste tussenresultaat. Het is echter belangrijk om de parameters van het simulated annealing algoritme zorgvuldig in te stellen. De kostfunctie die gebruikt wordt als indicatie voor de kwaliteit van de oplossing kan deze ambiguïties meestal detecteren. De kost van een configuratie waar flip ambiguïties optreden is namelijk hoger dan bij configuraties waar deze niet optreden. Desondanks kan het algoritme soms toch vast blijven zitten in een lokaal optimum. Dit duidt erop dat de parameters nog niet optimaal zijn ingesteld voor de verschillende configuraties. Dit probleem kan mogelijk verholpen worden door de parameters automatisch te configureren en dynamisch aan te passen door de tussenresultaten te analyseren.

In totaal werden 30 verschillende anchorconfiguraties getest. Deze kunnen worden onderverdeeld in 3 groepen, afhankelijk van het aantal vooraf bekende anchors. Dit varieerde van 3 tot 5. De spreiding van de resultaten neemt af naarmate het aantal anchors toeneemt. Soms bestaat er echter een groot verschil tussen de anchorconfiguraties onderling. Om dit verschil te verklaren werden de volgende 3 metrieken bestudeerd: anchorspreiding, anchorconnectiviteit en GDOP. Het lijkt erop dat de GDOP-waarde de beste indicator is in vergelijking met de spreiding en de connectiviteit van de anchors. Een nadeel van deze metrieken is dat ze nauwelijks rekening houden met de geometrie en de positie van de onbekende anchors, terwijl dit misschien een doorslaggevende factor kan zijn. Er is echter meer onderzoek nodig waarbij nieuwe metrieken moeten worden onderzocht aan de hand van verschillende testomgevingen.

Bibliografie

- [1] M. Ridolfi, A. Kaya, R. Berkvens, M. Weyn, W. Joseph, and E. D. Poorter, “Self-calibration and collaborative localization for uwb positioning systems: A survey and future research directions,” vol. 54, no. 4, 2021. [Online]. Available: <https://doi.org/10.1145/3448303>
- [2] K. Batstone, M. Oskarsson, and K. Åström, “Towards real-time time-of-arrival self-calibration using ultra-wideband anchors,” *2017 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2017*, vol. 2017-Janua, pp. 1–8, 2017.
- [3] M. Hamer and R. Dandrea, “Self-Calibrating Ultra-Wideband Network Supporting Multi-Robot Localization,” *IEEE Access*, vol. 6, pp. 22 292–22 304, 2018.
- [4] Q. Shi, S. Zhao, X. Cui, M. Lu, and M. Jia, “Anchor self-localization algorithm based on UWB ranging and inertial measurements,” *Tsinghua Science and Technology*, vol. 24, no. 6, pp. 728–737, 2019.
- [5] A. D. Preter, J. Anthonis, and J. Swevers, “Experiment Design for Ultra-wideband Sensor Node Calibration,” no. September, pp. 24–27, 2018.
- [6] W. Yu, X. Zhao, and G. Sun, “Self-calibration of anchor positions for indoor localization,” *2017 IEEE International Conference on Robotics and Biomimetics, ROBIO 2017*, vol. 2018-Janua, pp. 581–586, 2018.
- [7] C. M. Almansa, W. Shule, J. P. Queralta, and T. Westerlund, “Autocalibration of a mobile UWB localization system for Ad-Hoc multi-robot deployments in GNSS-denied environments,” *arXiv*, 2020.
- [8] M. Ridolfi, E. De Poorter, J. Fontaine, B. Van Herbruggen, W. Joseph, and J. Hoebeke, “UWB anchor nodes self-calibration in NLOS conditions : a machine learning and adaptive PHY error correction approach,” *Wireless Networks*, vol. 0, 2021. [Online]. Available: <https://doi.org/10.1007/s11276-021-02631-0>
- [9] K. Whitehouse, C. Karlof, A. Woo, F. Jiang, and D. Culler, “The Effects of Ranging Noise on Multihop Localization.”

- [10] Y. Liu and Z. Yang, *Location, localization, and localizability: Location-awareness technology for wireless networks*. Springer, 01 2011.
- [11] S. Pandey and S. Varma, “A Range Based Localization System in Multihop Wireless Sensor Networks: A Distributed Cooperative Approach,” *Wireless Personal Communications*, vol. 86, no. 2, pp. 615–634, 2016.
- [12] G. Mao, B. Fidan, and B. D. Anderson, “Wireless sensor network localization techniques,” *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.
- [13] E. Q. Shahra, T. R. Sheltami, and E. M. Shakshuki, “A comparative study of range-free and range-based localization protocols for Wireless Sensor Network: Using COOJA simulator,” *International Journal of Distributed Systems and Technologies*, vol. 8, no. 1, pp. 1–16, 2017.
- [14] F. Mekelleche and H. Haffaf, “Classification and comparison of range-based localization techniques in wireless sensor networks,” *Journal of Communications*, vol. 12, no. 4, pp. 221–227, 2017.
- [15] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less Low-Cost Outdoor Localization for Very Small Devices,” *Journal of Chemical Information and Modeling*, vol. 53, no. 9, pp. 1689–1699, 2013.
- [16] D. Niculescu and B. Nath, “Ad hoc positioning system (APS) using AOA,” *Proceedings - IEEE INFOCOM*, vol. 3, pp. 1734–1743, 2003.
- [17] Y. Shang and W. Ruml, “Improved MDS-based localization,” *Proceedings - IEEE INFOCOM*, vol. 4, no. June, pp. 2640–2651, 2004.
- [18] L. Doherty, K. Pister, and L. Ghaoui, “Convex position estimation in wireless sensor networks,” vol. 3, 02 2001, pp. 1655 – 1663 vol.3.
- [19] X. Ji and H. Zha, “Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling,” *Proceedings - IEEE INFOCOM*, vol. 4, pp. 2652–2661, 2004.
- [20] P. Biswas and Y. Ye, “Semidefinite programming for ad hoc wireless sensor network localization,” in *Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004*, 2004, pp. 46–54.
- [21] A. A. Kannan, G. Mao, and B. Vucetic, “Simulated annealing based localization in wireless sensor network,” *Proceedings - Conference on Local Computer Networks, LCN*, vol. 2005, no. March 2017, pp. 513–514, 2005.
- [22] C. Savarese, J. Rabaey, and K. Langendoen, “Robust positioning algorithms for distributed ad-hoc wireless sensor networks,” *Proceedings of the 2002 USENIX Annual Technical Conference*, no. May, 2002.

- [23] S. Čapkun, M. Hamdi, and J. P. Hubaux, “GPS-free positioning in mobile ad-hoc networks,” *Proceedings of the Hawaii International Conference on System Sciences*, vol. 00, no. c, p. 255, 2001.
- [24] A. Ihler, J. Fisher, R. Moses, and A. Willsky, “Nonparametric belief propagation for self-localization of sensor networks,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 809–819, 2005.
- [25] Y. Shang and W. Ruml, “Improved MDS-Based Localization,” *IEEE INFOCOM 2004*, vol. 4, pp. 2640–2651 vol.4, 2004.
- [26] G. MacGougan, K. O’Keefe, and R. Klukas, “Ultra-wideband ranging precision and accuracy,” *Measurement Science and Technology*, vol. 20, p. 095105, 07 2009.
- [27] J. Sidorenko, V. Schatz, N. Scherer-Negenborn, M. Arens, and U. Hugentobler, “Error corrections for ultrawideband ranging,” *IEEE Transactions on Instrumentation and Measurement*, vol. PP, pp. 1–1, 05 2020.
- [28] B. Hanssens, “Indoor ultra-wideband channel modeling and localization using multipath estimation algorithms,” Ph.D. dissertation, Ghent University, 2018.
- [29] S. Saunders and A. Aragón-Zavala, *Antennas and Propagation for Wireless Communication Systems*, 03 2007.
- [30] A. Kannan, B. Fidan, and G. Mao, “Analysis of flip ambiguities for robust sensor network localization,” *Vehicular Technology, IEEE Transactions on*, vol. 59, pp. 2057 – 2070, 06 2010.
- [31] P. Kaewprapha, J. Li, and N. Puttarak, “Network localization on unit disk graphs,” in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, 2011, pp. 1–5.
- [32] A. Franzin and T. Stützle, “Revisiting simulated annealing: A component-based analysis,” *Computers Operations Research*, vol. 104, 12 2018.
- [33] D. T. Connolly, “An improved annealing scheme for the QAP,” *European Journal of Operational Research*, vol. 46, no. 1, pp. 93–100, 1990.
- [34] H. Ishibuchi, S. Misaki, and H. Tanaka, “Modified simulated annealing algorithms for the flow shop sequencing problem,” *European Journal of Operational Research*, vol. 81, no. 2, pp. 388–398, 1995. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0377221793E0235P>
- [35] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953. [Online]. Available: <https://doi.org/10.1063/1.1699114>

- [36] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science (New York, N.Y.)*, vol. 220, pp. 671–80, 06 1983.
- [37] A. Coluccia and A. Fascista, "A review of advanced localization techniques for crowdsensing wireless sensor networks," *Sensors (Switzerland)*, vol. 19, no. 5, 2019.
- [38] A. Mesmoudi, M. Feham, and N. Labraoui, "Wireless sensor networks localization algorithms: A comprehensive survey," 12 2013.
- [39] B. Van Herbruggen, B. Jooris, J. Rossey, M. Ridolfi, N. Macoir, Q. Van den Brande, S. Lemey, and E. De Poorter, "Wi-pos: A low-cost, open source ultra-wideband (uwb) hardware platform with long range sub-ghz backbone," *Sensors*, vol. 19, no. 7, 2019. [Online]. Available: <https://www.mdpi.com/1424-8220/19/7/1548>
- [40] S. Geng, S. Ranvier, X. Zhao, J. Kivinen, and P. Vainikainen, "Multipath propagation characterization of ultra-wide band indoor radio channels," *2005 IEEE International Conference on Ultra-Wideband*, pp. 11–15, 2005.
- [41] M. Ridolfi, S. Van de Velde, H. Steendam, and E. De Poorter, "Analysis of the scalability of uwb indoor localization solutions for high user densities," *SENSORS*, vol. 18, no. 6, 2018. [Online]. Available: <http://dx.doi.org/10.3390/s18061875>
- [42] N. Macoir, J. Bauwens, B. Jooris, B. Van Herbruggen, J. Rossey, J. Hoebeke, and E. De Poorter, "Uwb localization with battery-powered wireless backbone for drone-based inventory management," *SENSORS*, vol. 19, no. 3, pp. 467:1–467:18, 2019. [Online]. Available: <http://dx.doi.org/10.3390/s19030467>
- [43] J. Fontaine, M. Ridolfi, B. Van Herbruggen, A. Shahid, and E. De Poorter, "Edge inference for uwb ranging error correction using autoencoders," *IEEE Access*, vol. 8, pp. 139 143–139 155, 2020.
- [44] S. Severi, G. Abreu, and D. Dardari, "A quantitative comparison of multihop localization algorithms," 04 2010, pp. 200 – 205.
- [45] N. Lasla, M. Younis, A. Ouadjaout, and N. Badache, "On optimal anchor placement for efficient area-based localization in wireless networks," *IEEE International Conference on Communications*, vol. 2015-Septe, pp. 3257–3262, 2015.
- [46] T. Kunz and B. Tatham, "Localization in wireless sensor networks and anchor placement," *Journal of Sensor and Actuator Networks*, vol. 1, no. 1, pp. 36–58, 2012.
- [47] S. Du, B. Huang, B. Jia, and W. Li, "Optimal anchor placement for localization in large-scale wireless sensor networks," *Proceedings - 21st IEEE International Conference on High Performance Computing and Communications, 17th IEEE International Conference on Smart*

- City and 5th IEEE International Conference on Data Science and Systems, HPCC/Smart-City/DSS 2019*, pp. 2409–2415, 2019.
- [48] G. Dudek and M. Jenkin, *Inertial Sensors, GPS, and Odometry*, 01 2008, pp. 477–490.
- [49] A. Al Baidhani and A. Alsudani, “Anchor selection by geometric dilution of precision for an indoor positioning system using ultra-wide band technology,” *IET Wireless Sensor Systems*, vol. 11, 12 2020.
- [50] D. Weyland, “Simulated annealing, its parameter settings and the longest common subsequence problem,” 01 2008, pp. 803–810.
- [51] S. L. Goh, G. Kendall, and N. Sabar, “Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem,” *Journal of the Operational Research Society*, vol. 70, pp. 1–16, 06 2018.

Appendices

Bijlage A: Localization Error

