Jonathan Beckhaus

Passive TDOA Localization in Multi Hop Scenarios using Ultra Wide Band Communication.

FAKULTÄT FÜR
INFORMATIK

Intelligent Cooperative Systems
Computational Intelligence

Passive TDOA Localization in Multi Hop Scenarios using Ultra Wide Band Communication.

Master Thesis

Jonathan Beckhaus

February 28, 2022

Supervisor: Prof. Dr.-Ing. habil. Sanaz Mostaghim

Advisor: Sebastian Mai

Jonathan Beckhaus: *Passive TDOA Localization in Multi Hop Scenarios using Ultra Wide Band Communication.*
Otto-von-Guericke Universität
Intelligent Cooperative Systems
Computational Intelligence
Magdeburg, 2022.

Abstract

In this thesis, a passive TDOA approach for ultra wide band localization is extended to be used in multi hop scenarios to enable a more flexible positioning system for large amounts of nodes. An error model is designed and used with measured clock drifts of DWM1000 modules in simulation to evaluate the capabilities of different proposed approaches. Experiments show that a combination of the proposed solver variance metric and hop count metric is able to significantly increase the amount of nodes localized in certain multi hop scenarios while maintaining similar median localization errors of around one meter as the compared single hop approach. It has furthermore been found that with different settings, proposed approaches were able to locate all nodes in an area four times the size of the area the original anchor nodes are placed in with only slightly higher median errors of around 1.3 meters.

Preface

Thanks to Nico, Max, Ruben and Tatjana for their support and Basti for his supervision.

Contents

List of Figures VII

List of Signs IX

1 Introduction 1
- 1.1 Tasks and Requirements . . . 1
- 1.2 Structure of the Thesis . . . 2

2 Basics 3
- 2.1 Two Way Ranging . . . 3
- 2.2 Time Difference of Arrival . . . 4
- 2.3 Gradient Descent Solver . . . 4
- 2.4 Clock Drift . . . 5
- 2.5 Multi Hop Localization . . . 6

3 State of the Art 7

4 Method 11
- 4.1 Passive TDOA . . . 11
- 4.2 Error Model . . . 13
- 4.3 Simulation Procedure . . . 20
- 4.4 Algorithms . . . 21
  - 4.4.1 Metrics . . . 21
  - 4.4.2 Anchor Selection . . . 24
  - 4.4.3 Anchor Decision . . . 25

5 Experiments and Results 27
- 5.1 Determining Clock Drift . . . 27

5.2 Error Model Evaluation 31
5.3 Step Size Determination 32
5.4 Metric Evaluation Variance Solver 35
5.5 Multi Hop Experiments 37
5.5.1 Multi Hop Approach Comparison 37
5.5.2 Analysis of Anchor Decision Thresholds 43
5.5.3 Influence of the Maximum Number of Allocated Anchor Nodes 48
5.5.4 Convergence Analysis 51
5.5.5 Performance on Different Scenarios 55

6 Conclusion **59**

7 Future Work **61**

Bibliography **63**

List of Figures

2.1 Two Way Ranging [18] 3
2.2 Gradient Descent Visualization [28] 5

4.1 Example visualizing a Position Dependency Graph (G_{PD}) and a Synchronization Tree (G_S) of the set of given nodes and their positions in the Geometric Position Graph (G_{GP}) 13
4.2 Example Synchronization Tree. Nodes selected as reference by tag i are marked in red. Node 3 is the root node used for the error calculation of the selected reference nodes 16
4.3 Example schedule with four anchor nodes A1 to A4 and one tag T with the resulting time differences of arrival marked as red arrows 18
4.4 Illustration of the angles of three anchor nodes A, B and C to each other around the tag T in two different scenarios 23

5.1 Sequence diagram of messages sent between two modules illustrating the setup used to determine the clock drift 28
5.2 Illustration of determined absolute clock drifts in ppm of three pairs of modules tested (displayed as green, blue and orange line) after warm up 29
5.3 Clock drifts of the three tested module pairs and their distribution 30
5.4 Average localization errors of 31 runs at iteration 250 and 500 using two different clock drift generation approaches 32
5.5 Average localization error over 500 iterations using different step sizes 33

5.6 Average localization error of 31 runs for different step sizes tested at iteration 250 35
5.7 Correlation of the variance solver metric with the localization error in a single hop scenario over 500 iterations with three different queue sizes 36
5.8 Visualization of the area setup of the nodes and original anchor nodes 38
5.9 Number of not localized nodes in a comparison table of different multi hop allocation and selection algorithm combinations . . . 42
5.10 Median error over all nodes in meters in a comparison table of different multi hop allocation and selection algorithm combinations 42
5.11 Median localization error in meters and number of not localized nodes using different thresholds for the **variance solver** metric as anchor decision algorithm (median over 31 runs) 46
5.12 Median localization error in meters and number of not localized nodes using different thresholds for the **variance angle** metric as anchor decision algorithm (median over 31 runs) 47
5.13 Distribution of median localization error in meters and number of not localized nodes at iteration 2000 for different maximum anchor values (k) for the selection functions 49
5.14 Convergence of the number of not localized nodes and the median localization error in meter for variance solver - hop count . 52
5.15 Convergence of the number of not localized nodes and the median localization error in meter for variance angle - hop count . 53
5.16 Scaling analysis of selected approaches showing median localization error in meter and not localized nodes over different areas and amount of nodes. 56

List of Signs

Graphs

G_{GP}	Geometric Position Graph
G_{PD}	Position Dependency Graph
G_S	Synchronization Tree

Multi Hop Algorithms

$s_{angle}(i)$	variance angle metric score of node i
$s_{solver}(i)$	variance solver metric score of node i
$\#_{hop}(i)$	hop count metric score of node i
$V_{reference}(i)$	set of reference nodes used by node i
$Q_n(i)$	set of the last n calculated positions of node i
$M_{angles}(i)$	set of clockwise angles of reference nodes to each other around node i

Error Model

$\theta(t)_i$	time of a clock i at a given reference time t
e_i^0	static time offset error of clock i
e_i^{drift}	clock drift induced error of clock i
t_i^{sync}	time between two synchronization messages to clock i

Δt_i^{sync}	time passed since the last synchronization messages to clock i
t_{sl}	length of a time slot in the sending schedule
d_i	drift of clock i
v_{signal}	traveling speed of the signals
e_i^{mh}	accumulated error of node i along a path to the root node r
$e_{i,j}^{tag}$	error induced by the tags clock drift from selected anchor j to tag i
$e_{i,j}^{pos}$	difference in assumed distance and actual distance between two synchronizing nodes i and j
$e_{j,i}^{mp}$	multi path error of the message sent from node j to node i
P_i^*	calculated position of node i
P_i	actual position of node i
$S_{SA}(i)$	set of selected anchor nodes by node i
$S(i)$	set of sending nodes around node i
$p_s(j)$	position of node j in a schedule
$TOF_{i,j}$	the time of flight between node i and j
$TOF_{i,j}^{err}$	the time of flight between node i and j with the corresponding errors
$TDOA_{i,j}^{err}$	time difference of arrival from the signal of anchor j at tag i with modeled errors and respect to other selected anchors.

1 Introduction

From mobile indoor robots to wireless sensor networks, more and more applications require localization. Often it is not possible to use GPS due to indoor application, usage in GPS denied environments such as woods and places close to buildings [40], or higher precision requirements.

One popular method to overcome this problem is to use ultra wide band (UWB) communication to determine a position [41, 27, 40]. There are different methods to set up UWB modules as a positioning system. Using a passive time difference of arrival (TDOA) setup enables a theoretically unlimited amount of nodes to be localized.

In this thesis, such a passive TDOA localization approach is extended to be used in multi hop scenarios, aiming to increase area coverage and flexibility while maintaining a simple setup. The system's performance is further analyzed in simulation.

1.1 Tasks and Requirements

One of the main questions to be answered in this thesis is whether it is possible to extend the passive TDOA approach to be used in multi hop scenarios, enabling nodes to share their calculated position estimate.

To evaluate such an approach, a simulation with an appropriate error model is to be implemented. With the simulation, the performance of such a multi hop approach can then be analyzed. Furthermore, it is to be found out, how well this approach localizes nodes within a single hop reachable area. Can it reduce the median localization error compared to a single hop approach ?

Typical GPS enabled phones reach an accuracy of about fife meters [44], with high precision GPS receivers reaching accuracies of under two meters [16]. To be able to achieve similar performance, a requirement for the system is that

the median localization error has to be below one meter. This also roughly matches the performance of other currently researched approaches of localization in wireless sensor networks [46] and of Wi-Fi and Bluetooth based indoor localization systems [21, 43, 15, 47].

With this constraint, how much more nodes can be localized with a multi hop approach compared with the single hop scheme in different scenarios and what are the limits of how far the range can be extended with such?

The research questions can be summarized as follows:

Q1 Is it possible to extend the passive TDOA approach into a multi hop scenario ?

Q2 Is applying multi hop strategies onto a single hop problem a possibility to improve the localization performance of the nodes ?

Q3 With a median localization error of below one meter, how many nodes can be localized with a multi hop approach ?

Q4 What are the limits of such a multi hop approach with the one-meter constraint ?

1.2 Structure of the Thesis

The structure of the thesis can be summarized as follows: After the **introduction** and the stated **tasks and requirements**, a brief overview of the **basics**, the work relates to, is given. Next, different research related topics are touched on in the **state of the art** chapter, followed by a detailed explanation of the theoretical considerations and **methods** used in this work. After that, the **experiments and** their **results** are presented. The thesis closes with a **conclusion** and considerations for **future work**.

2 Basics

In this chapter, a brief explanation of different localization principles, clock drift and the general idea behind multi hop localization is given.

2.1 Two Way Ranging

Two way ranging (TWR) is an approach to obtain a time of flight and thereby a distance between two ultra wide band capable nodes. Usually, double sided two way ranging is performed to compensate for a clock drift between the two modules. It consists of a sequence of messages sent between the nodes as in figure 2.1.

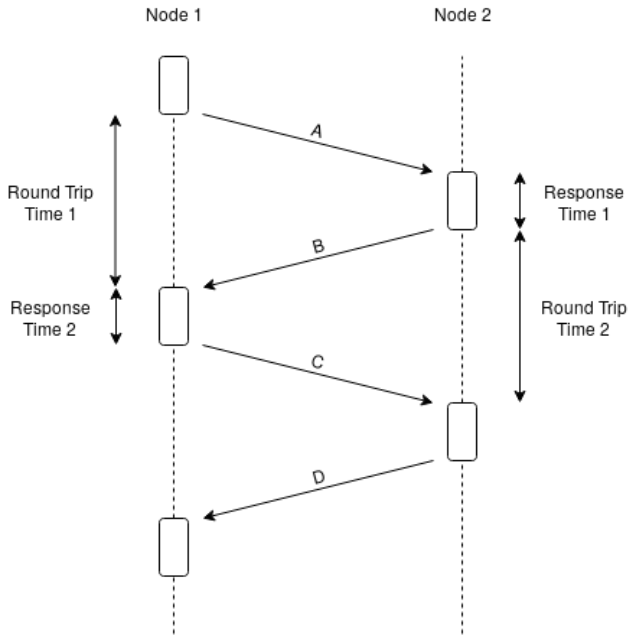

Figure 2.1: Two Way Ranging [18]

For each transmission, the previous registered local time stamps of sending and receiving are written into the message. With this, the round trip times and response times can be calculated. Subtracting the *response time (rpt)* from the *round trip time (rtt)* results in two times the time of flight. With this, the time of flight (TOF) of the two rounds can be calculated as follows:

$$TOF = [(rtt1 - rpt1) + (rtt2 - rpt2)]/4$$

The last transmission (D) is then used to transmit the resulting time of flight from Node 2 to Node 1. With a known speed of the signal (speed of light in the atmosphere for the application using ultra wide band signals), the time of flight can then be used to determine a distance between the two nodes. [18, 13, 9, 26]

2.2 Time Difference of Arrival

A time difference of arrival (TDOA) is the time between two receiving timestamps. This is used in an equally named localization principle, where multiple synchronized base stations receive a message from a sender.

Depending on the position of the sender relative to the base stations, the different stations receive the same signal at different timestamps. The further away a receiver is from the sender, the later the signal is received compared to a second receiver close to the sender, resulting in a higher time difference of arrival between the two receivers.

The differences of the receiving timestamps on the different base stations can then be used to calculate the position of the sender. [36, 18]

2.3 Gradient Descent Solver

There are many approaches to compute a position given different distances from tag to anchor nodes or different time differences of arrival between such base stations. While it is possible to solve these problems analytically, this often is not feasible due to errors in the input data.

More common approaches look at this problem as an optimization problem, solving it numerically by minimizing the error between the distances or time

differences of arrival generated by an estimated position and the actual input data. This can be done by computing the gradient over this error, iteratively "stepping" towards positions estimates minimizing this error, as visualized in figure 2.2. In the end, the position with the lowest error equals the best match of the input data to a position.

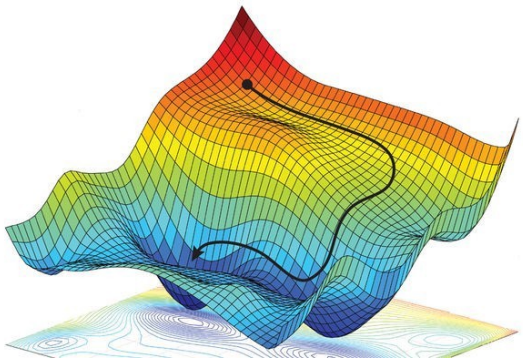

Figure 2.2: Gradient Descent Visualization [28]

A typical problem of such a gradient descent solver is, that depending on the error landscape and the step width in which the solver moves along the gradient, the solver might converge towards local optima or shows a diverging behavior. [33, 28, 17]

2.4 Clock Drift

Clock drift occurs between two or more independent clocks which run at different speeds. When the clocks are started simultaneously, they will "drift apart" and slowly desynchronize over time. The clock drift between two clocks is usually given in parts per million (ppm) with clock drifts of typical crystal oscillators used in normal wristwatches about 25 ppm [7].

If the exact drift of a clock C to a reference clock RC is known and does not change over time, it can be theoretically compensated by dividing the resulting time of that clock C by the drift of C plus one ($t_{RC} = t_C/(1 + drift_C)$). In reality however, the drift of a commonly used quartz driven clock depends on external factors such as temperature, resulting in different drift over time [37].

2.5 Multi Hop Localization

In classical radio based localization, base stations are used as reference to locate nodes in range. When these nodes then act as reference nodes or base stations to locate other nodes based on their calculated position, one speaks of multi hop localization. This can be used to locate nodes which are out of range of the initial base stations, or to assist in the localization of nodes which are in range. With each node in the localization chain between the original base stations and the node to be localized, the hop count is increased. A major disadvantage of this approach is the position error of each node being added up with each hop. [29]

3 State of the Art

One of the most common localization methods we encounter in our daily lives is GPS. It is broadly used in navigation systems for cars and built in to most smartphones these days. While the precision of typical GPS receivers is only about five meters [44, 8], one of the biggest advantages is its coverage around the world. While it is possible to increase the precision of GPS using an additional ground station, these D-GPS systems are very costly, have a high setup time of about 20 to 30 minutes and can only cover a certain area [4, 30]. Furthermore, blocking and reflecting of GPS signals can drastically decrease the localization performance of GPS, which makes it difficult to locate receivers in environments such as forests or places very close to buildings [40] and completely unfeasible inside buildings.

Various methods are used to obtain positions in such GPS denied environments. Camera based systems present one approach to localize objects in such areas. One of the most common methods for camera based indoor localization is using markers on the targets, which are then detected by permanently installed cameras in the room [20, 11, 35, 2]. A simple single camera localization system can be constructed using a webcam mounted on the ceiling detecting aruco markers on the tracking targets [25, 20]. Localization precision of under one centimeter when using a marker side length of 96 mm were achieved in [20] with such an approach covering an area of roughly two by two meters. The covered area is limited by the size of the markers and resolution of the camera used, often resulting in small localization areas. Commercially available multi camera systems such as the Vicon system, which uses infrared cameras and reflective markers, are able to locate the position of multiple targets with an accuracy of below one mm. [11, 35] The downside of these systems is their high price and their complicated, usually permanent setup. An overview and further comparison of different camera based localization systems, their performance and cost is given in [34].

Autonomous driving service and industry robots are usually equipped with laser scanners [12, 3]. These enable robots to locate themselves in a known environment, by tracking features of walls and obstacles. With odometry information of wheels and other sensors such as an inertial measurement unit, it is possible to perform simultaneously localization and mapping (slam). There, a map of the environment the robot is placed in is incrementally constructed, while the robot simultaneously determines its position within this map [24]. This approach enables the robot to locate itself in an unknown indoor environment and "provide[s] the means to make a robot truly autonomous" [24]. Accuracies of laser scanners are typically in centimeter range [5, 1]. Assuming a perfect slam algorithm, localization performance can be expected to be similar. Industry grade laser scanners however usually cost well above 1000 euros with cheaper hobby grade sensors starting at around 200 euros [14]. Additionally, enough computational power is needed for the execution of slam algorithms to obtain a position estimate with a reasonable update rate, making this approach a less viable option for larger amounts of robots to be tracked.

An other option to enable localization inside buildings are wireless transmission based systems. There, Wi-Fi and Bluetooth based solutions rely on commonly available technologies and enable a 2d localization accuracy of around one meter [47, 10, 15]. These localization approaches are often based on a received signal strength indicator (rssi) to estimate distances from setup base stations to the modules to be located [19, 23, 15]. With these distances, triangulation can then be performed to determine a position in 2d space. It is also possible to use the time of arrival and angle of arrival of signals to determine a position in a Wi-Fi based localization system [47]. Similar techniques can be used to determine positions with ultra wide band (UWB) communication. UWB based systems achieve a much higher localization precision of around 50 cm and below in 3d space [27, 42] while still being affordable with UWB communication capable hardware such as the DWM1000 modules being priced at less than 20 euros [6].

Kempke et al. [27] implemented a single sided two way ranging based indoor localization system using ultra wide band communication. They were able to track the path of a Parrot AR Drone with 39 cm median error using their proposed approach. Two way ranging usually requires at least two to four messages sent, to determine a distance between two UWB nodes. The system proposed by Kempke et al. [27] uses three antennas on the tag. With three antennas for each anchor and three different RF channels used, the proposed

system acquires 27 range estimates to each anchor node. Kempke et al. [27] were able to reduce the amount of sent packets to conduct these 27 range estimates to 31 with their custom ranging protocol.

Steup et al. [40] proposed a similar system using double sided two way ranging to localize an UWB tag. Their proposed system showed a median precision of around 0.2 meters with eight planar setup anchor nodes when compared to a high precision optical Vicon localization system. The computation of the position in the conducted outdoor experiment with the fully integrated system was entirely done on a microcontroller in the UWB nodes enabling a decentralized operation. With the ranging implementation used by Steup et al. [40], five packets are transmitted for one range estimate between two nodes. While this is significantly lower compared to [27], the scalability of this system is limited because with each added anchor or tag, the amount of sent messages drastically increases.

To tackle this problem, Tiemann et al. [41] proposed a TDOA based approach. There, a tag broadcasts a single message to be received by multiple synchronized anchor nodes. The different times of the message arriving at the anchor nodes, are then used to calculate the time differences of arrival and with these a position estimate. This is done on a server wired to all anchor nodes. In the approach proposed by Tiemann et al. [41], the synchronization is done by an additional synchronization node broadcasting repeating messages. While this system theoretically only requires one message sent by each tag for each localization, additional messages from the server to the localization targets are required so e.g. a mobile robot, mentioned by Tiemann et al. [41] as a possible application, can use the determined position for indoor navigation.

Wang et al. [45] proposed a TDOA based UWB localization approach, where the anchor nodes emit a scheduled signal. This enables the tags to locate themselves completely passively, theoretically enabling as unlimited amount of tags to be located without increasing the messages sent. The synchronization of the anchor nodes is done within the already transmitted signals of the schedule. Wang et al. [45] evaluated their system with four reference nodes and one tag and showed that it is capable of 3d indoor localization. With the passive localization of tags, no extra synchronization nodes required and the capability of computing the positions decentralized on the tags, this proposed approach has the potential to be extended into a multi hop capable localization system and is therefore further on used in this thesis.

4 Method

The following chapter explains the **passive multi hop** localization approach and its multi hop extension of this work. First, the general idea of the passive TDOA localization scheme is explained. This if followed by a more detailed explanation of the **error model** used in the further course of this thesis. Furthermore, the **simulation procedure** is elucidated. In the end, algorithms necessary for multi hop localization are presented.

4.1 Passive TDOA

In classical TDOA localization, the object to be tracked (tag) emits a signal received by the synchronized base stations (anchors) at different times. These are then used to calculate the different time differences of arrival necessary to calculate a position estimate. [36]

In the approach based on Wang et al. [45] the anchors emit a signal and the tag(s) only listens. This enables the position estimate to be calculated directly on the tags, completely passive. However, nodes need a certain amount of time to receive a message, and thereby cannot receive multiple messages at the same time or within a certain time frame. To avoid this, Wang et al. [45] proposed a schedule in which the anchor nodes emit their signal. With the length of the time slot known (or the time when the message has been sent), the tag can then use the times stamps it received the messages at, to calculate time differences of arrival and with them its position estimate. The big advantage of this approach is, that the amount of messages sent only depends on the amount of anchor nodes set up and is independent of the number of tags localizing themselves. This approach is further referenced as **passive TDOA**.

To enable passive TDOA in a multi hop scenario, a third node type next to the original anchor nodes and tags is introduced. While **anchors** (also referred to as original anchors) are the manually placed nodes with a known position

emitting localization signals, tag-anchors are tags that also emit a localization signal. As tags, tag-anchors also locate themselves using other anchors. To share their position and participate in the sending schedule, they need to be in sync with the other anchors. This results in a Synchronization Tree starting at one predefined master anchor node. Furthermore, every tag and tag-anchor node has a set of nodes it uses to calculate its position, creating a Position Dependency Graph.

Figure 4.1a shows an example setup of nodes at their positions. The Geometric Position Graph (G_{GP}) encodes the position and neighborhood. There are three original anchor nodes A0, A1 and A2, two tag-anchors AT0 and AT1, and one tag T in this example. The lines between the nodes visualize possible connections, limited due to a given transmission range. In this example, the tag T is not able to receive signals emitted by the anchors A0 and A1.

The Position Dependency Graph (G_{PD}) in figure 4.1b shows for each node which nodes they use as a reference to get their position. The position of the anchors is known, and therefore they form the "base" of this graph. This graph only shows one possible allocation of anchor nodes to nodes. For example, AT0 could additionally use AT1 to determine its position. However, in order to determine a two dimensional position, a node needs at least three reference nodes. Tag T can only receive one anchor node (A2) and therefore makes use of the two tag-anchors AT0 and AT1 to determine its position.

Furthermore, a Synchronization Tree (G_S) of the sending nodes is shown in figure 4.1c. Anchor node A0 is the synchronization master for this example. Since all sending nodes can listen to the synchronization messages of this node, the tree is rather flat. Again, this is only one solution for a Synchronization Tree in this scenario. Even though it would increase the synchronization hops, tag-anchor AT1 could also use AT0 as synchronization reference. However, expanding the network may result in tag-anchors not being able to directly receive signals from the synchronization master and instead having to use other nodes to synchronize themselves. Tags do not appear in this graph because they do not need to be in sync with the other anchor nodes. They only use the localization messages to compensate their clock drift.

In this work, the Synchronization Tree is always built so that each node uses the node with the lowest synchronization hop count available. If multiple of such nodes exist, the node with the lowest ID is always chosen. Furthermore, the original anchor node with ID 0 is always the synchronization master.

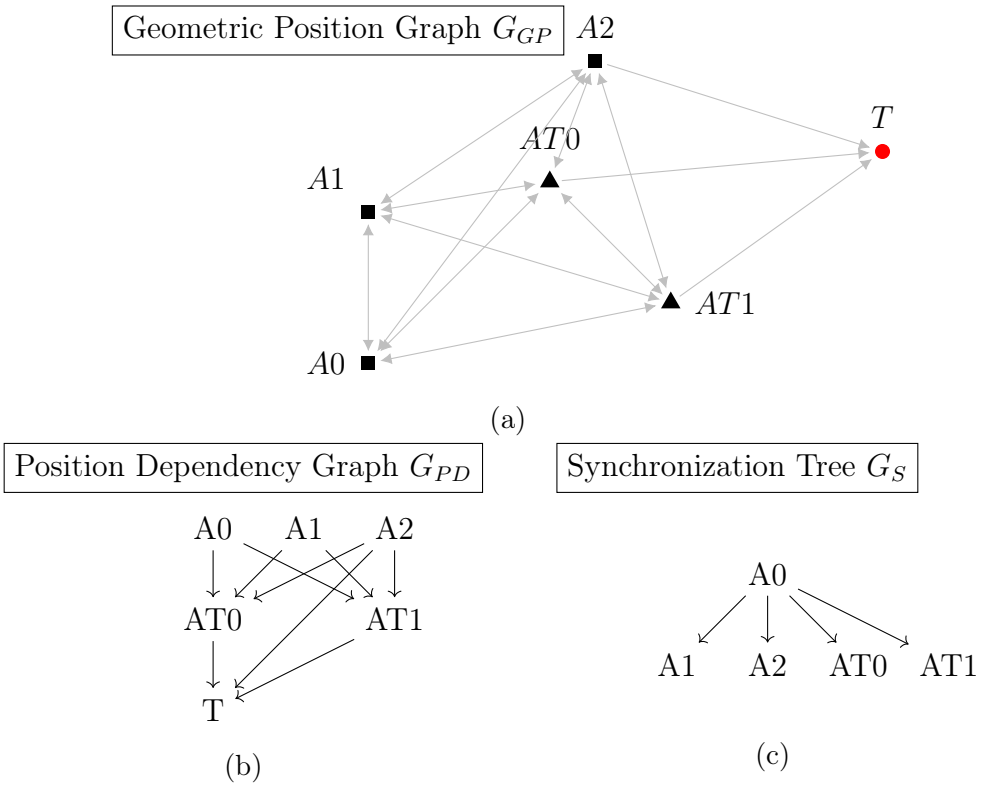

Figure 4.1: Example visualizing a Position Dependency Graph (G_{PD}) and a Synchronization Tree (G_S) of the set of given nodes and their positions in the Geometric Position Graph (G_{GP})

4.2 Error Model

In this chapter, a time model of the individual nodes is established. With this time model and the previously described multi hop procedure, an error model is then built with respect to correlation of time and calculated position as well as aspects of time synchronization.

The time of each anchor and tag-anchor node is modeled as follows:

$$\theta(t)_i = t + e_i \tag{4.1}$$

$$e_i = e_i^0 + e_i^{drift} \tag{4.2}$$

Each node is modeled to have an independent clock which has an offset and a drift. The time of a clock i ($\theta(t)_i$) at a given reference time t consist of that reference time t and an error e_i (equation 4.1). The error e_i is split up into two parts, a static offset error e_i^0 and an error due to clock drift e_i^{drift}. The clock

drift error for a specific time is the time passed since the last synchronization of the clock Δt_i^{sync} (in the reference time domain) times the drift of node i d_i in parts per million (equation 4.3).

$$e_i^{drift} = \Delta t_i^{sync} \cdot d_i \tag{4.3}$$

The model assumes that the clock drift is constant since the last time of synchronization. When a clock drift compensation between the reference clock and the module clock i is done, d_i equals the remaining drift after that compensation. Ideally, such a compensation would reduce the clock drift to zero, eliminating the clock drift and thereby the error induced by the drift. However, in reality clock drift compensation mechanisms cannot achieve this. This is mainly because the clock drift is not constant and often depends on external factors, as for instance the temperature of quartz oscillators. The remaining (much smaller) drift is then part of the error and therefore used for the error calculation.

The static error e_i^0 is introduced by the wireless synchronization of the nodes. With the speed the signal travels between two nodes, and the exact distance known, the time offset of one node to another can be compensated by sending a message from one module to the other containing a local time stamp. This synchronization is assumed to be exact if the positions of the two nodes is known. This is the case for the synchronization between two original anchor nodes. Therefore, the static error e_0 is set to zero in such a case. For tag-anchor nodes however, this does not hold, since they use their calculated position as reference. If the calculated positions differ from the actual position for one or two synchronizing nodes, this affects the distance used to account for the time the signal travels in the synchronization. This then leads to a time offset of the synchronized node to the node used for synchronization resulting in a static offset error e_0. For the case of time synchronization between two nodes with at least one being a tag-anchor node, the error e_0 can be calculated as follows:

$$e_i^0 = \frac{e_{i,j}^{pos}}{v_{signal}} \tag{4.4}$$

$e_{i,j}^{pos}$ is the difference in the assumed distance from node i to j to the actual distance between the two synchronizing nodes. With the speed of the signal (v_{signal}), the time offset can then be calculated as in equation 4.4. In the case

of the ultra wide band communication v_{signal} can be assumed to be equal to the speed of light in the atmosphere.

$$e_{i,j}^{pos} = |\overrightarrow{P_i^* P_j^*}| - |\overrightarrow{P_i P_j}| \tag{4.5}$$

Equation 4.5 shows how $e_{i,j}^{pos}$ is calculated. The distance between the node with the position P_i and its synchronization reference node with the position P_j equals the length of the vector $\overrightarrow{P_i P_j}$. The length of the vector between the estimated positions ($\overrightarrow{P_i^* P_j^*}$) is the distance used for the simple single message synchronization. In case of an anchor node, the calculated position P^* equals the actual position of the node, assuming perfect placement and measurements during setup. The difference of these two distances then equals the distance error $e_{i,j}^{pos}$, forming the foundation of the static offset error (equation 4.5). If the calculated positions of the two synchronizing nodes are equally offset, or one position estimate is offset in such away that it does not affect the calculated distance between the two nodes, it is possible that this has no effect on the synchronization resulting in $e_{pos} = 0$ even though the calculated positions have errors.

In a multi hop scenario the synchronization dependencies are given by the previously introduced Synchronization Tree G_S. Beginning by a synchronization master node, the static error e_0 and drift based error is inherited over each hop.

The time error of an anchor or tag-anchor node i is the sum of the errors of the nodes along the path to the root synchronization node r (equation 4.6).

$$e_i^{mh} = \sum_{j \in path_{i,r}} (e_j) \tag{4.6}$$

It is assumed that all sending nodes around the node i send in a repeating schedule one after another with time slot length of t_{sl} seconds (section 4.1). A sending node in the neighborhood of node i is a node in the Geometric Position Graph with an edge to node i. Given that one of the sending nodes is the synchronization node for node i, the time between the synchronization messages t_i^{sync} equals the length of the schedule time slots times the amount of sending nodes in the neighborhood $len(S(i))$, as in equation 4.7. t_i^{sync} therefore is the maximum time since the last synchronization. This is used for Δt_i^{sync}

to calculate the e^{drift} part (equation 4.3) of the error accumulated in equation 4.6. Since the schedule is not further specified and hence the times for the synchronization of the nodes along the path to the root node are not known, the worst case is always used by setting $\Delta t_i^{sync} = t^{sync}$ and it is ensured that the errors are not assumed to be too low.

$$t_i^{sync} = t_{sl} \cdot len(S(i)) \tag{4.7}$$

When a node i determines its position, it selects sending nodes in its neighborhood which are notated as $S_{SA}(i)$. Figure 4.2 shows a Synchronization Tree where the red nodes (8, 9, 10, 11) are the selected reference nodes by a tag i. For localizing tag i, the anchor nodes need to be synchronized to each other. However, it is not necessary to calculate the e^{mh} for the selected anchor (tag-anchor) nodes until root node 0 of the Synchronization Tree, because they all already share node 3 as a root node. The error introduced from the synchronization from node 0 to node 3 affects all selected anchor nodes equally and hence has no influence on the synchronization between the selected anchor nodes. When the error e_j^{mh} is calculated for all nodes j in a set of selected anchor nodes $S_{SA}(i)$ by node i ($j \in S_{SA}(i)$), the common root node of these selected nodes is used as root node r in equation 4.6. The resulting error value is further referred to as $e_j^{mh}(S_{SA}(i)), j \in S_{SA}(i)$.

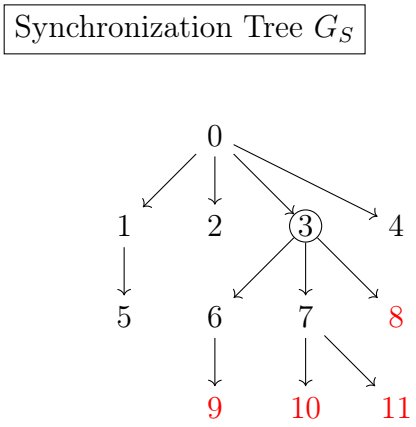

Figure 4.2: Example Synchronization Tree. Nodes selected as reference by tag i are marked in red. Node 3 is the root node used for the error calculation of the selected reference nodes

In addition to the errors of the selected anchor or tag-anchor nodes, an error based on the clock drift of the tag is modeled. Similar to the error calculation of

the anchor nodes, the nodes around the tag are assumed to send in a repeating schedule one after another, again with a time slot length of t_{sl}. The schedule is assumed to have the selected nodes send first (in an undefined order), followed by all the other nodes in range.

Figure 4.3 shows an example schedule for four anchor nodes A1 to A4 selected by tag T ($S_{SA}(T) = [A1, A2, A3, A4]$). The schedule begins with node A1 sending. With t_{sl} known (e.g. through listening to the sending time stamps of the anchor nodes), the tag T can then compute the time differences of arrivals (TDOA) of the four nodes (red arrows in figure 4.3). The resulting time differences of arrival are relative to the first received signal, hence the resulting time difference of arrival for the first node (A1 in case of the example in figure 4.3) always equals zero.

However, this example assumes that the clock of the tag T has no drift compared to the synchronized selected anchor nodes. With T having such a clock drift, t_{sl} used for the calculation in the time domain of T, does not equal t_{sl} in the time domain of the reference nodes anymore. The error based on this depends on the length of the schedule time slots (t_{sl}), the clock drift of the tag d_{tag} and it scales with the position of the nodes in the schedule. This is modeled in equation 4.8 where $p_s(j)$ is the position of reference node j in the schedule and d_i the clock drift of node i. When a clock drift compensation is done, d_i equals the remaining clock drift. Because no complete schedule is established for the simulation, the assignment of selected anchor nodes to a position in the schedule is done at random for each set of selected anchor nodes $S_{SA}(i)$. The resulting error $e^{tag}_{i,j}$ needs to be considered for each selected anchor node j in respect to all selected anchor nodes in $S_{SA}(i)$ for tag i.

$$e^{tag}_{i,j}(S_{SA}(i)) = (p_s(j) - 1) * t_{sl} * d_i \tag{4.8}$$

To calculate the time differences of arrival with the two previous introduced types of errors $e^{mh}_j(S_{SA}(i))$ and $e^{tag}_{i,j}(S_{SA}(i))$, the errors are added to the time of flight from the selected reference nodes to the tag first (equation 4.9). Equation 4.11 shows how the time differences of arrival are then yielded. This is done by subtracting the time of flight (TOF) of the first node in the schedule (j=1) from all the times of flight. This results in the TDOA being relative to this

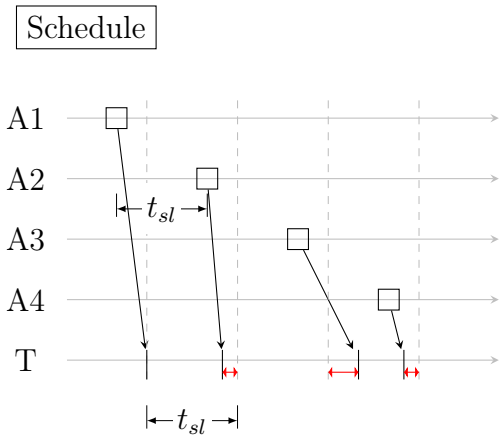

Figure 4.3: Example schedule with four anchor nodes A1 to A4 and one tag T with the resulting time differences of arrival marked as red arrows

first node. Hence, the TDOA of the first reference node (j=1) equals zero (as in the example in figure 4.3).

$$TOF^{err}_{j,i}(S_{SA}(i)) = TOF_{j,i} + e^{tag}_{i,j}(S_{SA}(i)) + e^{mh}_{j}(S_{SA}(i)) \tag{4.9}$$

$$TOF_{j,i} = \frac{|\overrightarrow{P_j P_i}|}{v_{signal}} \tag{4.10}$$

$$TDOA^{err}_{j,i}(S_{SA}(i)) = TOF^{err}_{j,i}(S_{SA}(i)) - TOF^{err}_{1,i}(S_{SA}(i)) \tag{4.11}$$

The time differences of arrival $TDOA^{err}_{j,i}(S_{SA}(i))$ of all reference nodes j in the set of selected reference nodes $S_{SA}(i)$ of tag i can then be used to calculate a position estimate of node i. These time differences include two types of error, one based on the Synchronization Tree, another based on the clock drift of the receiving tag.

Signals sent with ultra wide band communication are subjected to multi path errors. Reflections of the signal result in different longer paths of the signal from the sender to the receiver. When the message on the direct path is

received, other reflected signals representing that same message which would be received later are ignored. However, if the direct path is blocked, or the signal gets lost, signals over another path may be received. This results in a different time of arrival of the message at the receiving node, interfering with the localization scheme. To model this, a multi path error $e_{j,i}^{mp}$ is applied to the time of flight in equation 4.11 as in equation 4.12.

$$TDOA_{j,i}^{err}(S_{SA}(i)) = TOF_{j,i}^{err}(S_{SA}(i)) \cdot e_{j,i}^{mp} - TOF_{1,i}^{err}(S_{SA}(i)) \cdot e_{1,i}^{mp} \quad (4.12)$$

If the signal on the direct path from node j to i arrives, the multi path error $e_{j,i}^{mp}$ is set to 1. In case of the DWM1000 module, the user manual claims a 97% of transmissions to succeed with a total air utilization of <18% [32]. This probability is used to decide, whether the direct path signal arrives (Algorithm 1, line 6). If this direct transmission fails, there is a chance of a multi path signal to arrive with a length of 1.2 times the direct path ($e_{j,i}^{mp} = 1.2$). When this fails, two more paths with $e_{j,i}^{mp} = 1.4$ and $e_{j,i}^{mp} = 1.6$ each have a chance (Algorithm 1, line 8-13). The probability for each of the three multi paths is set to 5%, resulting in a combined probability of approximately 14% probability of getting one of these multi path errors if the direct path signal does not arrive.

Algorithm 1 multi path error

1: $e_{j,i}^{mp}$
2: $Rne \leftarrow random(0,1)$
3: $Rmp1 \leftarrow random(0,1)$
4: $Rmp2 \leftarrow random(0,1)$
5: $Rmp3 \leftarrow random(0,1)$
6: **if** $Rne \leq 0.97$ **then**
7: $\quad e_{j,i}^{mp} \leftarrow 1$
8: **else if** $Rmp1 \leq 0.05$ **then**
9: $\quad e_{j,i}^{mp} \leftarrow 1.2$
10: **else if** $Rmp2 \leq 0.05$ **then**
11: $\quad e_{j,i}^{mp} \leftarrow 1.4$
12: **else if** $Rmp3 \leq 0.05$ **then**
13: $\quad e_{j,i}^{mp} \leftarrow 1.6$
14: **else**
15: $\quad e_{j,i}^{mp} \leftarrow None$
16: **end if**

If none of the three multi paths were chosen, $e_{j,i}^{mp}$ is set to $None$, declaring a lost transmission. There are three options how a lost transmission can be handled. First, the node can use the last calculated time difference of arrival of the node with the lost transmission. However, this does not work if the set of allocated nodes $S_{SA}(i)$ has changed. The second would be that the receiving node can calculate its position without the lost signal, only using all other signals received. This can only be done if at least three other messages were received when calculating a two dimensional position. The third option is to wait for the next transmission and not calculate the position in this period if one signal is lost. With an expected value of approximately 2.5% for a lost message, the frequency in which no position can be calculated using the last option is reasonable. Because of this and the simplicity of this solution, the third option is further used. Overall, this rather simple model with the three defined multipliers enables the inclusion of the multi path error without modeling an environment and the physics of signals.

With the time differences of arrival calculated as described above, a position is then calculated using a simple iterative gradient descent solver as the base approach in [33]. In the implementation of this work, the parameter *step size* defines how much of the calculated gradient is applied to the previous guess each iteration. The solver has further more been modified to reset its current guess to a randomly sampled position, if the guess leaves the area the nodes are distributed over by more than 10 meters in any direction to avoid runaway.

4.3 Simulation Procedure

A simulation is set up to in two dimensional space. It consists of two phases: a **calculation phase** and an **allocation phase**.

In the **allocation phase**, first all nodes can decide whether they want to share their position, by becoming a tag-anchor or stop sharing their position and to change back to being a tag. Then they decide which of the available anchors in range they want to use to determine their position in the following calculation phase. The original anchors are not subjected by the allocation phase and always emit a signal. In the end of the allocation phase, the Position Dependency Graph G_{PD} and the Synchronization Tree G_S are updated.

During the **calculation phase**, nodes use the signals from the selected anchors to determine their position estimate. To do so, the errors are calculated as

explained above, using the actual Position Dependency Graph G_{PD} and the Synchronization Tree G_S. For each node and each calculation step, the errors of the anchor nodes are recursively calculated until their common root node in G_S and the last calculated positions of the selected anchor nodes are used to determine the position of the corresponding tag or tag-anchor. The order in which the positions of the nodes are calculated is random but consistent over the simulation, depending on the randomly assigned IDs of the nodes. The calculation phase performs **one** position update per node. However, the whole phase is repeated certain times, set by the simulation parameter *"updates per allocation"*. Once all repetitions of the calculation phase are finished, the next allocation phase is started and the whole process is repeated until a certain amount of calculation updates (iterations) have been performed.

The implementation of the simulation and with it the error model and all the following algorithms was done using Python with typical libraries such as Numpy, Pandas and Scipy. For the evaluation, Matplotlib and Seaborn were additionally used to generate graphs and other visualizations.

4.4 Algorithms

In this chapter, different metrics to estimate the position quality of a node are proposed. Furthermore, different selection algorithms for choosing surrounding anchor nodes to determine a node's position, making use of the metrics, are shown. Second, different anchor decision algorithms are proposed which decide whether a node shares its position or not.

4.4.1 Metrics

To estimate the performance of a calculated position of a node, different metrics are presented in this chapter.

Hop Count

The hop count of a node i ($\#_{hop}(i)$) indicates over how many hops the position has been calculated. It is the maximum hop count out of every node used as reference for calculating the position of the node i plus one (equation 4.13).

$V_{reference}(i)$ is the set of reference nodes used. The hop count of an original anchor node is zero.

$$\#_{hop}(i) = max(\{\#_{hop}(x)|x \in V_{reference}(i)\}) + 1 \tag{4.13}$$

Over each hop, position and time synchronization errors are accumulated. It can thereby be assumed, that a node with a lower hop count has a better position estimate than a node with a higher hop count.

The implementation is not resistant to cycles. However, if there is such a case, that for example node A uses node B as reference to calculate its position and vice versa, the hop count of the involved nodes will increase with each update, penalizing the nodes.

Variance Solver

The position of a node is updated regularly. This metric calculates the variance of $Q_n(i)$, the set of the last n calculated positions of node i, coordinate wise (equation 4.14). The resulting score then equals the length of the vector containing the different variances.

$$s_{solver}(i) = |\begin{pmatrix}\sigma^2(Q_x^n(i))\\ \sigma^2(Q_y^n(i))\end{pmatrix}| \tag{4.14}$$

If the calculated positions barely differ over time, the solver has converged to a certain position. Even though the position it has converged to might be a local optimum and not close to the real position, it is assumed that the solver mostly converges to the actual position of the node. However, if the calculated positions drastically change over time, resulting in a high value of s_{solver}, the position estimate of the node is likely to be off. This can be caused by the solver not being converged yet, but also by high and inconsistent errors of the time differences used to calculate the position. When considering moving nodes, assuming a perfect tracking, slowly moving nodes would be preferred over fast moving nodes with this metric.

Variance Angle

The following metric rates the distribution of the selected reference nodes around the node. The score is lower if the anchor nodes used to calculate the

position of node i are distributed more evenly around node i. This is achieved by calculating the clockwise angles $M_{angles}(i)$ of the reference nodes to each other around the calculated position of the node. The score of node i $s_{angle}(i)$ then equals the variance of these angles (equation 4.15). Figure 4.4 shows an example with three anchor nodes A, B and C used by the tag T. α, β and γ are the angles between the reference nodes and form the set $M_{angles}(T)$. Figure 4.4-L illustrates a scenario where the anchor nodes are well distributed around the tag T resulting in equally sized angles and therefore in a low score s_{angle}. The scenario shown in figure 4.4-H however, represents a case where the angles between the nodes vary a lot resulting in a high score s_{angle}.

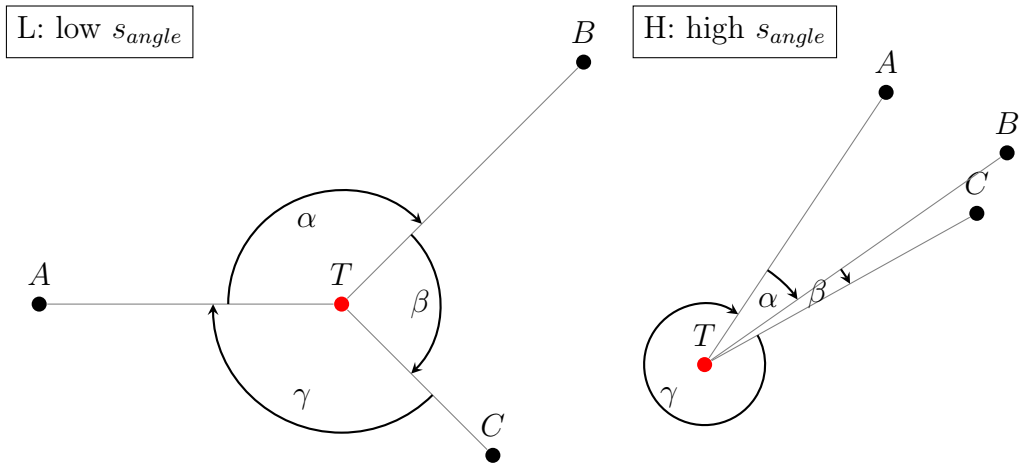

Figure 4.4: Illustration of the angles of three anchor nodes A, B and C to each other around the tag T in two different scenarios

$$s_{angle}(i) = \sigma^2(M_{angles}(i)) \tag{4.15}$$

The impact of an error in the measured time distances of arrival between anchor nodes and tag on the calculated position is lower when the reference nodes are distributed equally around the tag compared to them being clustered in a small "viewing angle" of the tag (e.g. figure 4.4-H). Since the score is calculated based on the current position estimate of the tag rather than its actual position, the score might only produce meaningful results if the calculated position of the tag is close to its actual position. However, if the calculated position of the tag is way off, it can be argued that this likely results in a higher score

s_{angle}. It is assumed that overall, nodes rated with a lower score s_{angle} will have a better position estimate than nodes with a higher score.

4.4.2 Anchor Selection

To calculate a position, a tag needs to choose which signals of the sending nodes in range it uses to calculate its position. It can either use only the signals emitted by the original **anchors**, or it can use **all** the available signals. Only choosing the setup anchor nodes ensures that only nodes with a perfect position estimate and low TDOA-errors are selected. However, doing so results in nodes not being able to localize themselves if they cannot receive at least tree anchor node signals because they are out of range.

Always using all sending nodes in range to calculate a position on the other hand may result in a bad position estimate by using tag-anchor nodes as reference with a high position or high synchronization error. The performance of this approach highly depends on the nodes in range which are emitting a localization signal.

Another way of deciding which anchors to use is ranking them by a metric and selecting the k best performing available nodes in that metric as reference nodes. The performance of this approach highly depends on the metric used. If the metric is able to correctly estimate the position and synchronization error of a node, only the nodes with the lowest errors are selected. The three metrics proposed above are used, resulting in tree different versions of the algorithm:

- take k lowest hop count,
- take k lowest variance solver,
- take k lowest variance angle

If nodes have the same score and cannot all be selected without exceeding the limit of k nodes, nodes of the same score are selected randomly. While this is unlikely to happen for the variance solver and the variance angle metric, the hop count has discrete values (number of hops $\#_{hop}$) which makes it probable that two or more nodes in range have the same value.

None of the proposed selection methods requires additional messages sent. This ensures that tags can still determine their position passively, only by listening. Necessary information, as the score of a node, can be transmitted

within the already sent localization messages of the emitting anchor (or tag-anchor) nodes.

4.4.3 Anchor Decision

To enable multi hop, tags need to share their position and become a tag-anchor. An intuitive way to decide if a node shares its position or not is to simply let **all** nodes share their position as soon as they have calculated one for themselves. While this enables the selection algorithms presented above to choose from all nodes, the amount of sent messages leads to a higher localization error. This is due to the increased schedule length resulting in fewer synchronizations and a higher impact of the remaining clock drift (section 4.2). However, restricting the emission of localization messages to anchors only would result in a single hop scenario, not letting the algorithms described above choose any but the original anchor nodes. Ideally, only nodes with a high position accuracy and a low synchronization error share their position. But since the actual errors cannot be determined by the nodes, the **variance solver** and the **variance angle** metric presented above are used to estimate the performance of the nodes. Two thresholds (t_{low}, t_{high}) are then used to determine when a node i starts sharing (metric(i) $< t_{low}$) and when a node stops sharing (metric(i) $> t_{high}$) its position. The hop count metric is not used because it is too discrete. Furthermore, it would result in many nodes sending in the area already covered by the original anchor nodes.

5 Experiments and Results

In the following chapter, different experiments are conducted and evaluated, to answer the previously stated research questions. First, the clock drift of DWM1000 UWB modules is determined. After some basic parameter settings are then examined in various single hop simulations, different multi hop approaches are analyzed.

5.1 Determining Clock Drift

The simulation of the passive TDOA localization is based on custom modules (as in [40]) containing a DWM1000 module for the ultra wide band communication, and an STM32 microcontroller to handle I/O and enable local position calculation. One big factor of the position accuracy in localization systems using DWM1000 modules is the clock drift of the modules [39, 22]. To generate meaningful results in simulation, the following experiment has been conducted to determine the clock drift of the actual modules.

Therefore, two modules have been set up in a distance of around one meter facing each other with the antenna side of the module. The modules were placed inside a box to ensure thermal stability, since crystal oscillators as installed in the modules are sensitive to change in temperature.

To determine the clock drift between two modules, one module periodically sends a message received by the second module. Figure 5.1 illustrates the messages sent between the two modules A and B in a sequence diagram. There, module A periodically sends messages at the timestamps t_{An} for the n^{th} message in its time domain t_A. Module B receives the n^{th} messages at t_{Bn} in its own time domain. Assuming that the speed in which the clocks of the two modules A and B run is the same and that the traveling time of the signal is always the same, the following equation can be concluded (equation 5.1).

$$t_{A,n+1} - t_{A,n} = t_{B_n+1} - t_{B_n} \tag{5.1}$$

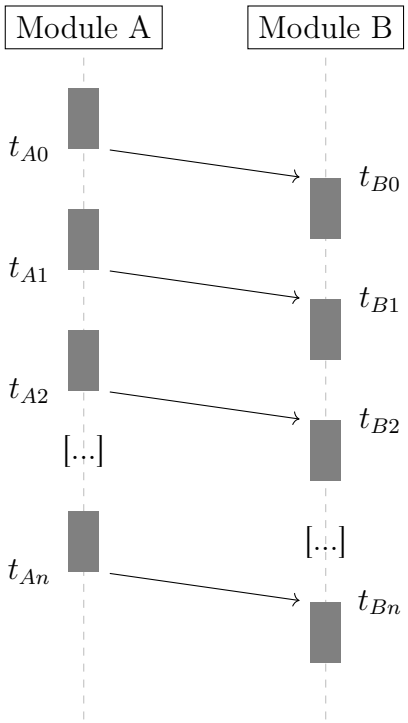

Figure 5.1: Sequence diagram of messages sent between two modules illustrating the setup used to determine the clock drift

Equation 5.1 states that the time between two messages sent equals the time between the same two messages received in the different time domains. Rearranging this leads to equation 5.2.

$$(t_{A,n+1} - t_{A,n}) - (t_{B_n+1} - t_{B_n}) = 0 \tag{5.2}$$

However, if the clocks of two modules do not run in the same speed, the difference of the time between the two messages sent on the one module and the time between the two messages received on the other (in the local times of the modules) equals the clock drift between the two modules as in equation 5.3.

$$(t_{A,n+1} - t_{A,n}) - (t_{B_n+1} - t_{B_n}) = d \tag{5.3}$$

During the experiment, three pairs of modules have been tested. For each pair, one module was set up to send a message five times a second, containing the local time of sending. The other module receives this messages and forwards the sending and receiving timestamps t_{An} and t_{Bn} to a computer via serial

interface (UART) where they are stored. The pairs of modules each ran about 30 minutes before the logging was started to ensure warmed up modules.

Figure 5.2 shows the calculated clock drifts of the three pairs of modules based on the recorded times over one hour. The resulting clock drifts are rather constant over the logging period, with clock drifts between two and 5.5 ppm. Decawave claims a factory calibration of the DWM1000 modules to less than 2ppm clock drift [31]. The clock drift measured during this experiment is between two modules, which puts the results roughly in line with the information provided in the data sheet of the DWM1000 [31].

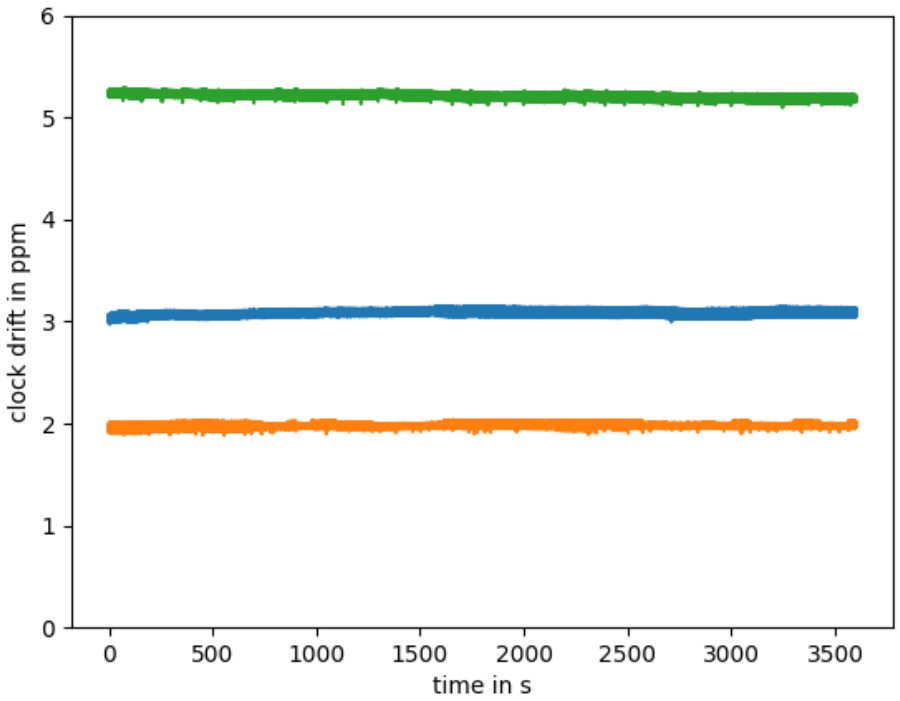

Figure 5.2: Illustration of determined absolute clock drifts in ppm of three pairs of modules tested (displayed as green, blue and orange line) after warm up

Using these clock drifts as input for the error model in section 4.2 would result in very high errors, rendering position calculation with the passive TDOA approach impossible. A clock drift of five ppm over one millisecond would

result in an error of five nanoseconds. The UWB signal traveling with the speed of light would travel about 1.5 m in that time. This can be solved by applying a drift compensation using the already sent messages of the anchor nodes.

Figure 5.3 enables a closer look at the drifts of the modules pairs (left) and their distribution (right). All the distributions are in a range of about 0.15 ppm, with a variance between 0.021 and 0.029 ppm. The second and third pairs show multiple repeating spikes. This might be caused due to multi path effects, noise in the medium (e.g. Wi-Fi) or the temperature compensation of the module.

To approximate a clock drift compensation, it is assumed that such a compensation would be able to determine the mean clock drift over the logged period of one hour. With that, the deviation of that mean value in the distribution can be used as drift d in the proposed error model.

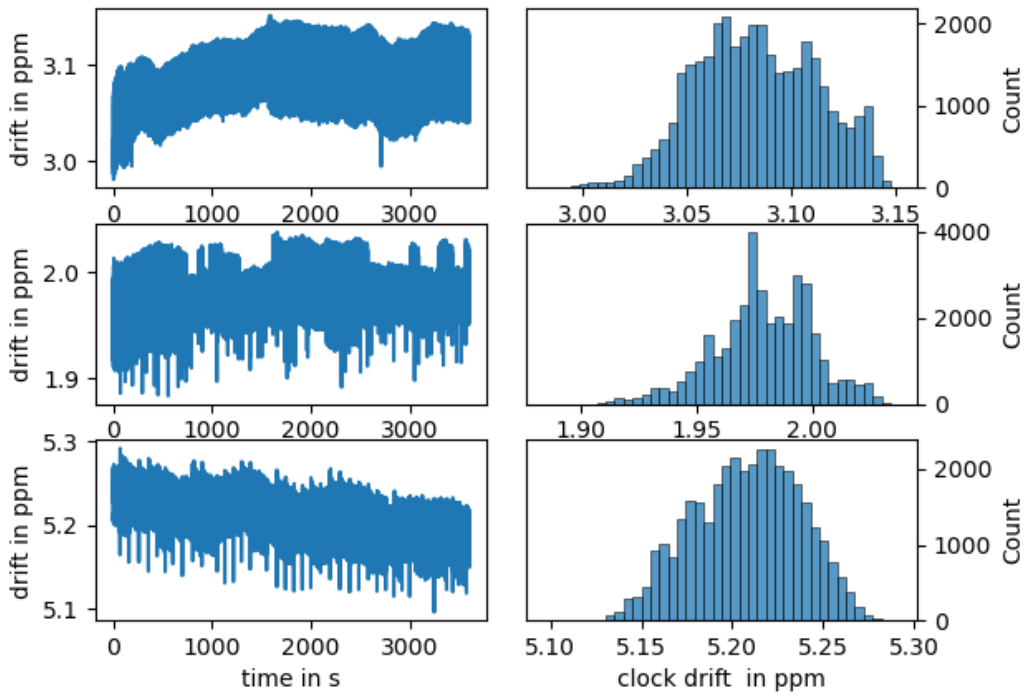

Figure 5.3: Clock drifts of the three tested module pairs and their distribution

5.2 Error Model Evaluation

With the following experiment, two different ways of generating clock drift values for the error model in a single hop scenario are compared. It is checked whether the used distribution of clock drifts can be approximated with a normal distribution.

The first approach for the generation of clock drift is based on the recorded values as introduced in section 5.1. In the simulation, nodes are randomly assigned to one of the three distributions. They then use the actual recorded values minus the mean of that distribution as drift d as input for the error model. This approach is further referenced as **module clock drift generation (module cdg)**.

The second approach approximates the distributions as normal distributions. The mean of these distributions is set to zero and the variance is randomly sampled between the min (0.0215 ppm) and max (0.0293 ppm) variance of the recorded distributions for each simulated node. Using a normal distribution enables a more general and more easily reproducible model. It is further referenced as **normal clock drift generation (normal cdg)** In both approaches, a time slot length of five milliseconds is used for the error generation ($t_{sl} = 5ms$).

To compare these two approaches, the following scenario has been simulated: Eight anchor nodes and 100 tags have been sampled on a two dimensional field with a size of 10x10m. The amount of anchor nodes chosen represents a number of nodes which is realistic to manually set up, as in Steup et al. [40] The transmission range has been set to 10 meters.

Tags were set up to only listen to anchor nodes and never to emit a signal by themselves, thereby describing a single hop scenario. The iterative solver was configured to use 0.25 as step size.

Both approaches to generate the drift d for the error model (section 4.2) have been run 31 times over 500 iterations each. In one iteration, each node performs one localization update by computing one solver step. The randomly generated node positions and initial solver guesses were kept the same for all the runs to eliminate other influences on the result but the different drift generation approaches.

Both approaches converge at about 200 iterations, with an average localization error of around 15 cm. After that, a slight increase in localization performance

until iteration 500 has been observed. Figure 5.4 visualizes the distribution of the average localization error at iteration 250 and 500 over the 31 runs per clock drift generation approach. The average position error of the nodes is distributed from about 11 cm to 18 cm for the 31 runs. Using the Mann–Whitney U test to compare the two methods at iteration 250 and 500, there is no significant difference between the two. (iteration 250: p=0.7675, iteration 500 p=0.9327)

Further on, the normal distribution approach is used as no significant difference to the module based drift generation could be found.

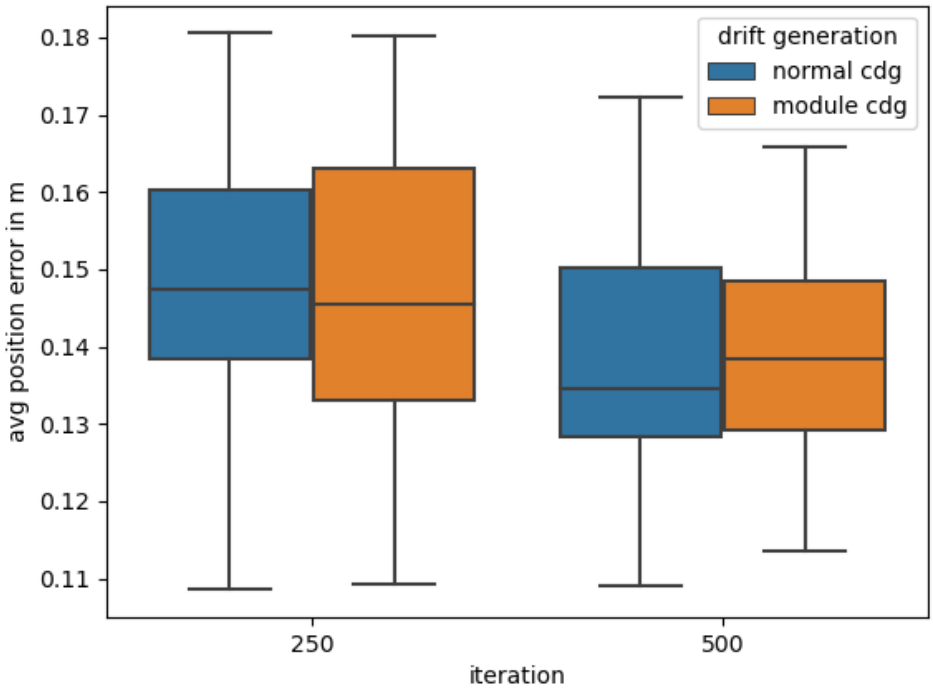

Figure 5.4: Average localization errors of 31 runs at iteration 250 and 500 using two different clock drift generation approaches

5.3 Step Size Determination

One critical parameter in this work is the step size of the gradient solver. The step size determines, how much the position estimate is "moved" along

the calculated gradient. In the following experiment, the influence of this parameter on the yielded solutions is analyzed.

Again, a simulation environment with eight randomly placed anchor nodes and 100 tags in a 10 by 10 meter environment with a transmission range of 10 meters has been used. Seven different step sizes have been tested in 31 runs, each over 500 iterations, with the nodes configured to perform single hop localization. All runs were performed over the same randomly generated node configuration with the same randomly generated solver starting positions to ensure a fair comparison. The normal distribution based drift generation approach has been used for the error model with five ms as schedule time slot size (section 4.2).

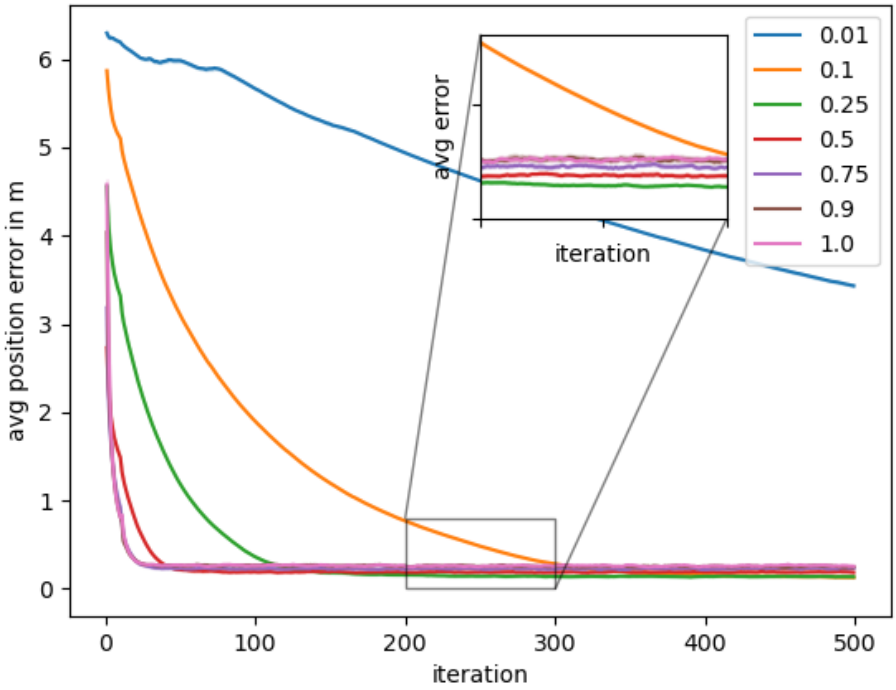

Figure 5.5: Average localization error over 500 iterations using different step sizes

Figure 5.5 shows the different convergence of the step sizes. The step sizes have been chosen to have a higher resolution at the lower and upper end of the scale. It can be seen that the larger the step size, the faster the convergence. Using a step size of 0.01, the runs do not converge over the 500 iterations the experiments were run, while all other runs converge within the simulated iterations. Between the step size of 0.75, 0.9 and 1.0 no convergence benefit is directly visible anymore. Furthermore, the figure 5.5 shows that the step size of 0.25 has a lower overall average position error once converged compared to higher step sizes.

Figure 5.6 visualizes the average position error of the 31 runs per step size at iteration 250. Since most of the runs have converged at this iteration and with respect to computational cost, results at this iteration are looked at in more detail. The runs performed with a steps size of 0.01 and 0.1 show a very high average error because they have not converged at this point. Furthermore, it is visible, that out of the converged runs, the higher step sizes produced higher average position errors. The runs with step size 0.25 significantly outperform all other runs. Using the Mann–Whitney U test, significance levels of lower than 10^{-9} were computed.

During the evaluation of the experiment, it has been noticed that the variance of the position errors of the 100 tags in the different runs increased with higher step sizes. At iteration 250, the mean variance of the position errors in the runs with a step size of 0.25 was approximately 0.5 cm, while the mean variance exceeds 1.5 cm for the tested step sizes greater than 0.5. This might be caused due to higher influences of short term errors in the time differences of arrival, resulting in a bigger "response" in the position updates when using higher step sizes.

Overall it can be seen that smaller step sizes result in a slower convergence but tend to outperform higher step sizes once converged. Further on, the step size of 0.25 is used in this work since it outperforms other solutions while converging in a reasonable amount of iterations.

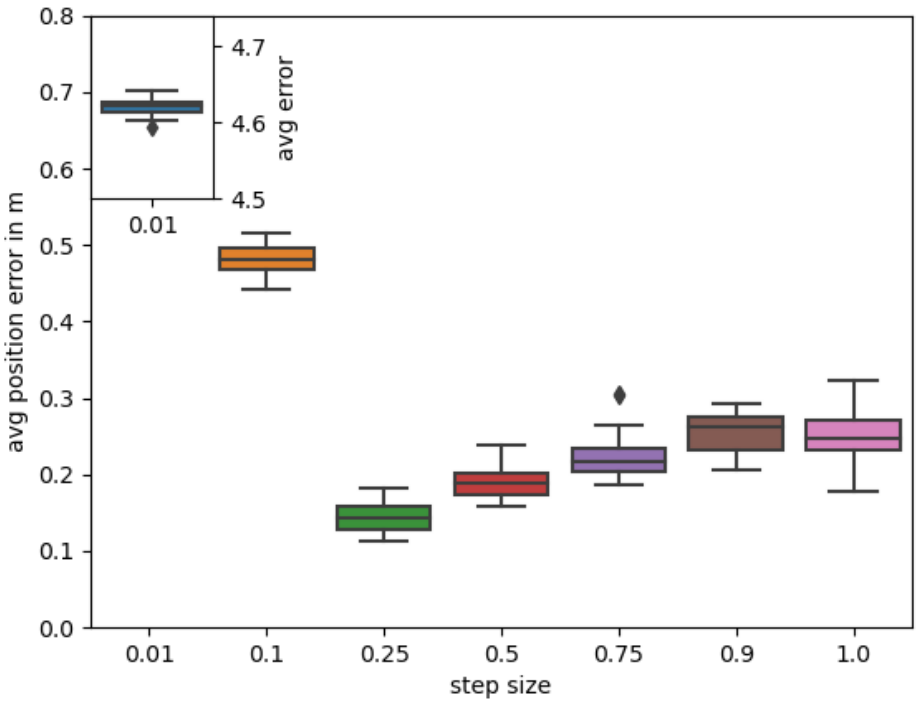

Figure 5.6: Average localization error of 31 runs for different step sizes tested at iteration 250

5.4 Metric Evaluation Variance Solver

The variance solver metric proposed in section 4.4.1 is based on the assumption, that a high variance over the last n calculated positions of a node i indicates a high position error and that a low variance, a low position error. To verify this assumption, a simulation with a randomly generated configuration of 100 tags and eight anchor nodes, within an area of 10 by 10 meters has been performed in a single hop scenario where tags only listen and determine their position. The transmission range has been set to 10 meters. For each node at each position update (iteration), the variance solver metric has been calculated over the experiment run of 500 iterations. Furthermore, three different queue sizes have been used to calculate the metric (10, 20 and 50). For each queue size 31 runs using the same initially randomly generated configuration of nodes, were performed. The starting positions of the solvers were randomly sampled for

each run. Figure 5.7 shows the correlation of the calculated metric with three different queue sizes to the position errors of the tags at each iteration using the Pearson correlation coefficient over all 31 runs per queue size.

It is clearly visible, that the correlation at the first 10 iterations is quite low. This is caused by the queue $(Q_n(i))$ over which the variance is calculated is being filled up during the first iterations. In these first iterations, there are too few position estimates calculated to allow the metric to work properly. However, once the first 20 iterations have passed, the correlation of all three queue sizes is increasing well over 0.7. The resulting correlation coefficients of over 0.8 at the end of the run and around 0.75 on average indicate an overall strong correlation between the metric and the actual position error [38].

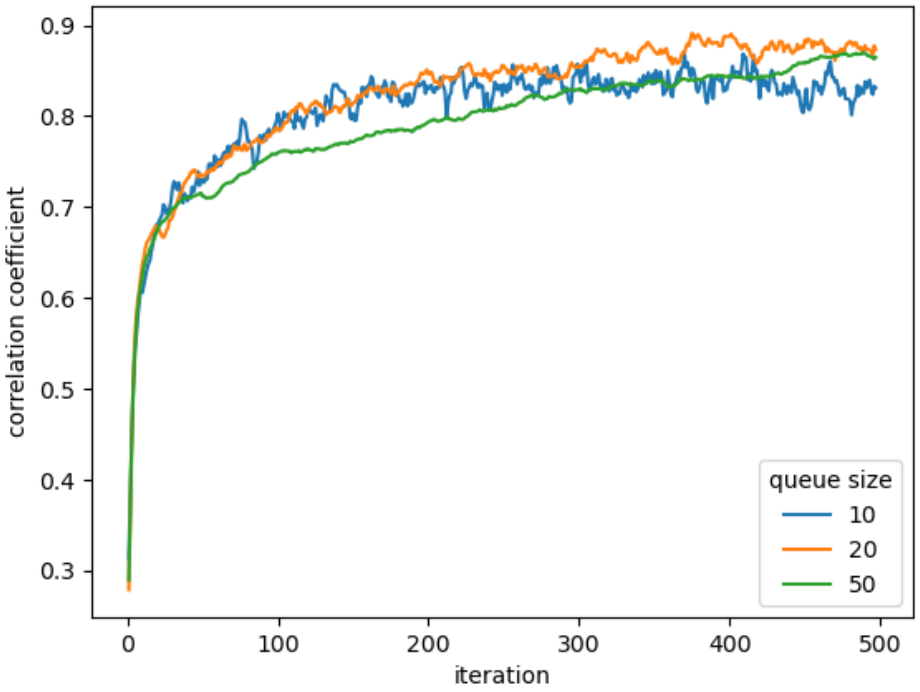

Figure 5.7: Correlation of the variance solver metric with the localization error in a single hop scenario over 500 iterations with three different queue sizes

Furthermore, it is visible that the correlation increases with the calculated positions converging. The correlation using the variance solver metric with

a queue size of 50 is more consistent than the lower queue sizes. However, the convergence of the correlation is slower. This is especially visible between iteration 100 and 200 and is caused by the big queue size acting as a low pass filter. Because of that, it cannot account for the rather dynamic change of the solver positions, especially during the first 200 iterations where most of the calculated positions have not fully converged.

Using a queue size of 20 results in the highest overall correlation for this experiment, while maintaining a good correlation in lower iterations. 20 is therefore used as queue size for this metric during the following experiments in this work.

5.5 Multi Hop Experiments

5.5.1 Multi Hop Approach Comparison

In the following experiment, a multi hop scenario is simulated comparing multiple combinations of anchor decision and anchor selection algorithms. Eight anchor nodes have been randomly placed on an area of 10 by 10 meters ($s_A = 10m$), while 100 nodes have been randomly sampled onto an area of 20 by 20 meters ($s_N = 20m$). Figure 5.8 visualizes the two areas. This specific configuration has been chosen because it represents a more difficult scenario than placing the anchor nodes in the middle of the map when maintaining the same amount of nodes and sizes of the area. Furthermore, this scenario contains a wide range of different situations. The nodes within the area A_A can usually be localized within one hop because they have enough original anchor nodes in range to do so, at the edge of the area only a few original anchors reach the nodes while nodes in the top right corner will not be able to receive any signals of the eight anchors.

During the experiment, different anchor selection and anchor decision algorithm combinations are tested in the described scenario. As anchor decision algorithms, the two proposed metric based approaches are tested in this experiment (**variance angle** and **variance solver**). For the variance angle decision algorithm a lower threshold t_{low} of 0.6793 and a higher threshold t_{high} of 1.1331 was set. The variance solver decision algorithm has been used with a single threshold of $t_{low} = t_{high} = 0.0215$. Besides these two approaches, setting up **all** nodes to share their calculated position as well as only letting the original

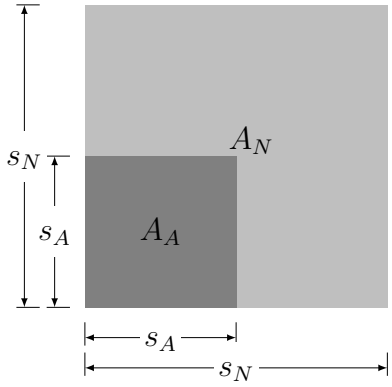

Figure 5.8: Visualization of the area setup of the nodes and original anchor nodes

anchor nodes share their position is investigated. The hop count metric is not used as anchor decision because of its discrete value and low resolution.

To decide which of the sending nodes should be used, the proposed algorithms in section 4.4.2 have been examined. The three metric based algorithms (**hop count**, **variance angle** and **variance solver**) are set up to use the eight best available options regarding their underlying metric ($k = 8$). Furthermore, the approach of always choosing **all** available nodes and only choosing the original **anchor** nodes were tested.

In addition to these algorithms, a combination of the metric based approaches (**comb**) has been run in the experiment. For the anchor decision, this combination is an equally weighted sum of the variance angle metric and the variance solver metric. There, the upper thresholds of the metrics are used to normalize them, as in equation 5.4. A single threshold of $t_{low} = t_{high} = 1$ is then used for the decision based on this metric.

$$s_{comb}^{decision}(i) = (\frac{s_{angle}(i)}{1.1331} + \frac{s_{solver}(i)}{0.0215})/2 \tag{5.4}$$

A similar approach has been used for the combination metric used as anchor selection algorithm. This additionally includes the hop count metric. To normalize this, the minimum hop count necessary to go along the diagonal of the map is used. On the 20 by 20 meter sized map with a transmission range of 10 meters, this results in around three hops. The combined score for the anchor selection is then calculated as in equation 5.5. Again, the eight best

performing nodes in range using that metric are selected as reference nodes when using this combined approach as anchor selection algorithm ($k = 8$).

$$s_{comb}^{selection}(i) = \frac{s_{angle}(i)}{1.1331} + \frac{s_{solver}(i)}{0.0215} + \frac{\#_{hop}(i)}{3} \tag{5.5}$$

All configurations have been run 31 times for 2000 iterations with an allocation phase each 25 solver position updates (*updates per allocation* = 25). In the end, 80 allocation phases have been performed over the 2000 iterations.

Figure 5.10 and 5.9 show the results of different combination of anchor decision and anchor selection algorithms in the described multi hop scenario at iteration 2000. While the heat map in figure 5.9 shows the number of not localized nodes, figure 5.10 visualizes the median localization error in meters over all nodes including the ones who have not obtained a position estimate. This metric has been chosen to better compare the performance regarding all nodes enabling the inclusion of not localized nodes. In both heat maps, each number represents the median out of the performed 31 runs per configuration.

Further on, the notation [x,y] is used to simplify references to specific entries in the heat map, where x notates an anchor decision algorithm and y an anchor selection algorithm e.g.[all, v-solver] for the lower left entry of the table.

In figure 5.9 it can immediately be seen, that all approaches using the anchor decision or anchor selection algorithm where only the original anchors may send or are chosen as reference produce the highest number of not localized nodes of 35. Only enabling the original anchor nodes to send or only selecting the original anchor nodes when calculating a position renders the scenario to single hop. Because the setup anchor nodes are only placed in a part of the map as in figure 5.8 and with the limit of the transmission range of 10 m, many nodes are not able to receive any signals and hence are unable to locate themselves.

Figure 5.10 shows that the median error when using only the original anchors as sending nodes is very consistent. There, the selection algorithms do not have much influence, since they will always use all the available sending nodes. This is because they either take all or the best eight nodes and there are only eight original anchors available. However, setting all nodes to share their position as soon as they have calculated one, but then only letting nodes

use the original anchors as reference ([all, anchor]) results in a high median error of 3.6 meters. Due to the high amount of sending nodes, the length of the synchronization schedules drastically increases, resulting in a much higher influence of the uncompensated clock drifts d_i (section 4.2). In addition to that, more messages sent results in more messages affected by the multi path model, influencing the quality of the localization.

When all nodes share their calculated positions, all anchor selection algorithms except the one only choosing original anchors ([all,anchor]) manage to locate all nodes (figure 5.9). However, the median localization error does differ significantly. When using all available nodes for calculating the positions when all nodes also share their position results in the highest median error of around 20 meters. There, loops in the Position Dependency Graph and the inclusion of bad position estimates are probably the main reasons for the bad performance. Furthermore, this approach requires substantially more computation time since the amount of reference nodes used is not limited. In contrast, selecting from the available nodes based on their hop count ([all, hop count]) manages a median position error of 2.3 meters while also locating all nodes.

Selecting the variance angle algorithm as anchor decision results in 26 and less localized nodes while maintaining similar median errors over all nodes as the original anchor based decision approach. It can be seen, that the approach using all available anchors in combination with the variance angle decision ([v-angle, all]) does not perform very well. While this has the highest number of localized nodes when using the variance angle metric for nodes to decide whether they share their position, it comes with a very high median position error of 19 meters which is a strongly dominated solution (e.g. by [all, hop count]).

The variance solver metric based decision algorithm reduces the amount of not localized nodes significantly compared to the v-angle based approach to around 18, at the cost of slightly higher median localization errors. The combination [v-solver, v-angle] is able to increase the amount of localized nodes even further but with a median error of 1.9 meters. This however is close to the 2.3 meters error of the [all, hop count] approach which manages to localize all nodes and therefore not really suitable. When looking at results with a median error of under one meter, using the variance solver decision algorithm in combination with the hop count as anchor selection [v-solver, hop count] has the highest number of localized nodes.

The combined approach overall performs very similar to the variance solver algorithm. This might indicate, that the proposed scaling of the underlying metrics fails to consider each metric evenly. The metric does not yield any benefit and combination in which the metric outperforms the variance solver approach in some way ([all, comb], [comb, v-angle]) are dominated by [all, hop count].

Summarizing the analysis the following points can be seen:

- The presented problem has two main objectives, the number of localized nodes and the median position error of all nodes
- All of the proposed multi hop combinations are able to locate more nodes than the single hop algorithms which only use the original anchors. Some of them do so while maintaining a similar median position error.
- When all nodes share their position and use all the available nodes to calculate their own position ([all,all]), the median error is highest.
- Of the approaches which manage to localize all nodes, the combination [all, hop count] has the lowest median localization error.
- Only considering solutions with a median error of below one meter, the combination [v-solver, hop count] is able to locate the most nodes.
- The proposed combined score does not yield a noticeable benefit.

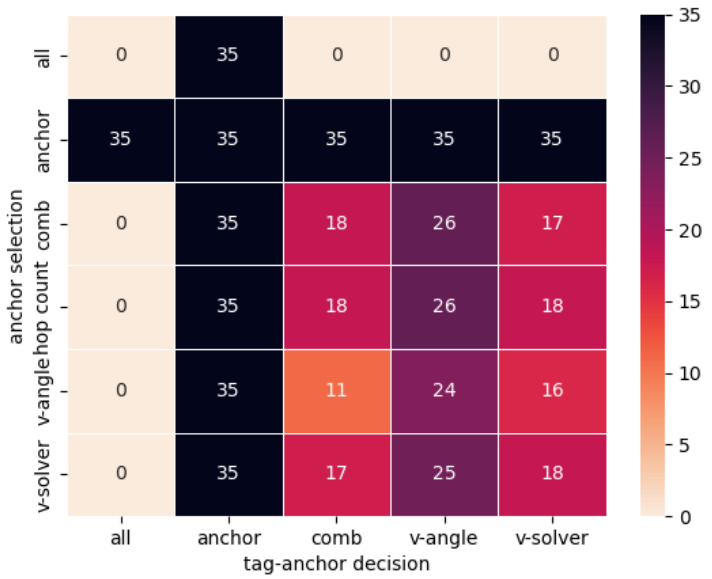

Figure 5.9: Number of not localized nodes in a comparison table of different multi hop allocation and selection algorithm combinations

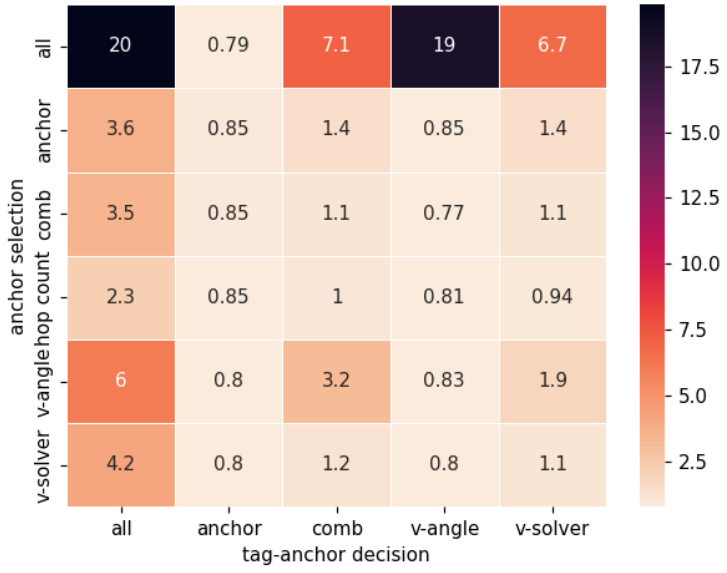

Figure 5.10: Median error over all nodes in meters in a comparison table of different multi hop allocation and selection algorithm combinations

5.5.2 Analysis of Anchor Decision Thresholds

The proposed variance angle and variance solver metrics are used with two thresholds in the anchor decision algorithm, where nodes decide whether they share their position or not. During this experiment, different configurations for the thresholds t_{low} and t_{high} are tested.

To do so, a multi hop scenario with eight defined anchor nodes randomly distributed over an area of 10 by 10 meters ($s_A = 10$) and 100 nodes randomly distributed over 20 by 20 meters ($s_N = 20$). A layout of the different areas can be seen in section 5.5.1 figure 5.8. Two different algorithm sets are tested, one using the variance solver, the other the variance angle metric as anchor decision algorithm. For both sets, the hop count based algorithm is used as anchor selection function. This approach has been chosen to ensure a fair comparison between the two decision algorithms in a multi hop scenario with reasonable outcomes. As experiment 5.5.1 has shown, using 'all' or 'anchor' would not be feasible. Using variance solver, variance angle or the combination approach as anchor selection bias free results cannot be ensured. The hop count anchor selection algorithm is used with a maximum number of anchors to select of eight ($k = 8$).

set	anchor selection	anchor decision
1	hop count	variance solver
2	hop count	variance angle

Table 5.1: Algorithm Sets

Combinations of 10 values logarithmically sampled for t_{low} and t_{high} were simulated, resulting in 100 configuration per algorithm set. For the variance angle based approach, the range of the tested parameters is set from 0.0877 to 8.7729, which is determined to be the maximum feasible value this metric can have. One worst case scenario for the variance angle metric would be a set of three referenced anchor nodes all at the same position resulting in three angles: zero, zero and 2*pi over which the variance is 8.7729. For the variance solver metric, the tested parameters range from 0.01 to 10. The sampling and scale is based on preliminary experiments with respect to the behavior of the metrics and computational cost of the experiments.

The scenario has been run 31 times for each configuration over 1250 iterations, with randomly sampled solver starting positions for each run. As in the

previous experiment, one allocation phase every 25 localization updates has been performed. The decreased run time of the experiments of 1250 instead of 2000 iterations compared to the previous experiment is due to computational constraints. The convergence behavior and the influence of the number of iterations if further analyzed in section 5.5.4.

Figure 5.11 and figure 5.12 show the resulting median localization errors and number of not localized nodes of the two metrics variance angle and variance solver at iteration 1250. In both figures, the values show the median over 31 runs.

It can be seen immediately that both metrics cover a wide range of solutions, strongly dependent on the parameters t_{low} and t_{high}. Higher threshold values result in more nodes localized while increasing the median localization error while lower threshold values achieve the opposite. In both approaches, increasing the threshold values beyond a certain point does not yield any benefit anymore, only increasing the median error when all nodes are already localized. For the variance solver metric this point can be found at 0.2154 for t_{low} and t_{high} and for the variance angle metric at 1.1331 for the two thresholds. Furthermore, it is visible, that the t_{low} has an overall bigger impact on the results than t_{high}. This is because nodes need to achieve t_{low} at least once to share their position. If a node does not manage to do so, t_{high} has no influence. When the lower threshold is set high enough, the influence of t_{high} increases since more nodes manage to surpass t_{low} at least for a short time.

With the variance solver metric as anchor decision, a threshold t_{low} of 0.0215 achieves a median position error of around one meter while localizing all but around 20 nodes (figure 5.11). The upper threshold t_{high} has no real influence on that result. Increasing the lower threshold to 0.0464 slightly increases the median localization error while significantly improving the amount of localized nodes. There, increasing t_{high} further increases the amount of localized nodes. However, this also results in a higher inter quantile range (IQR) of the number of not localized nodes over the 31 runs. This shows that a higher hysteria given through higher difference between t_{high} and t_{low} does localize more nodes in median but at the cost of more inconsistency.

The variance angle metric shows a similar behavior, although it is shifted towards higher thresholds (figure 5.12). Lower thresholds of 0.0877 and 0.1463 can barely be met by any node, effectively rendering the scenario single hop. Increasing t_{low} to 0.4072, the algorithm is able to locate 10 more nodes while

maintaining the same median localization performance of below 0.8 meters (e.g. $t_{low} = 0.4072, t_{high} = 0.6793$). Increasing t_{high} beyond 3.1528 results in inconsistent behavior and a high IQR of over 30 for not localized nodes. A steep decline in not localized nodes can be seen when setting t_{low} to 1.1331 or higher. Also increasing t_{high} to 1.1331 and higher with $t_{low} = 0.6793$ shows this behavior. This indicates that the chosen range of thresholds might not be sufficient in this range. Still, similar performance to the variance solver metric can be observed when comparing solutions localizing all nodes.

One of the research questions of this thesis is to analyze the capabilities of passive TDOA in multi hop with median localization errors below one meter. To do so, configurations of t_{low} and t_{high} with the highest amount of localized nodes below the one meter in median localization error were chosen based on this experiment and are further on used. For the variance solver based anchor selection algorithm, t_{low} and t_{high} are therefore set to 0.0215. The thresholds for the variance angle based approach were set to 0.6793 for t_{low} and 1.1331 for t_{high}. While this results in a surprisingly high number of localized nodes, still being well under the one-meter mark for the median localization error, this set of parameters comes with a rather high inter quantile range (IQR) for the number of not localized nodes, nearly three times the IQR of the variance solver metric with the chosen parameters.

With these thresholds, the proposed metrics have a powerful way to influence the solution generated in this multi objective problem. When considering solutions with slightly higher median localization errors than one meter, both approaches manage to drastically increase the amount of localized nodes, even managing to localize all nodes in the given scenario. Furthermore, it has been noticed that the variance solver metric as anchor decision algorithm achieved more consistent results overall with smaller inter quantile ranges over the performed 31 runs per configuration compared to the variance angle approach.

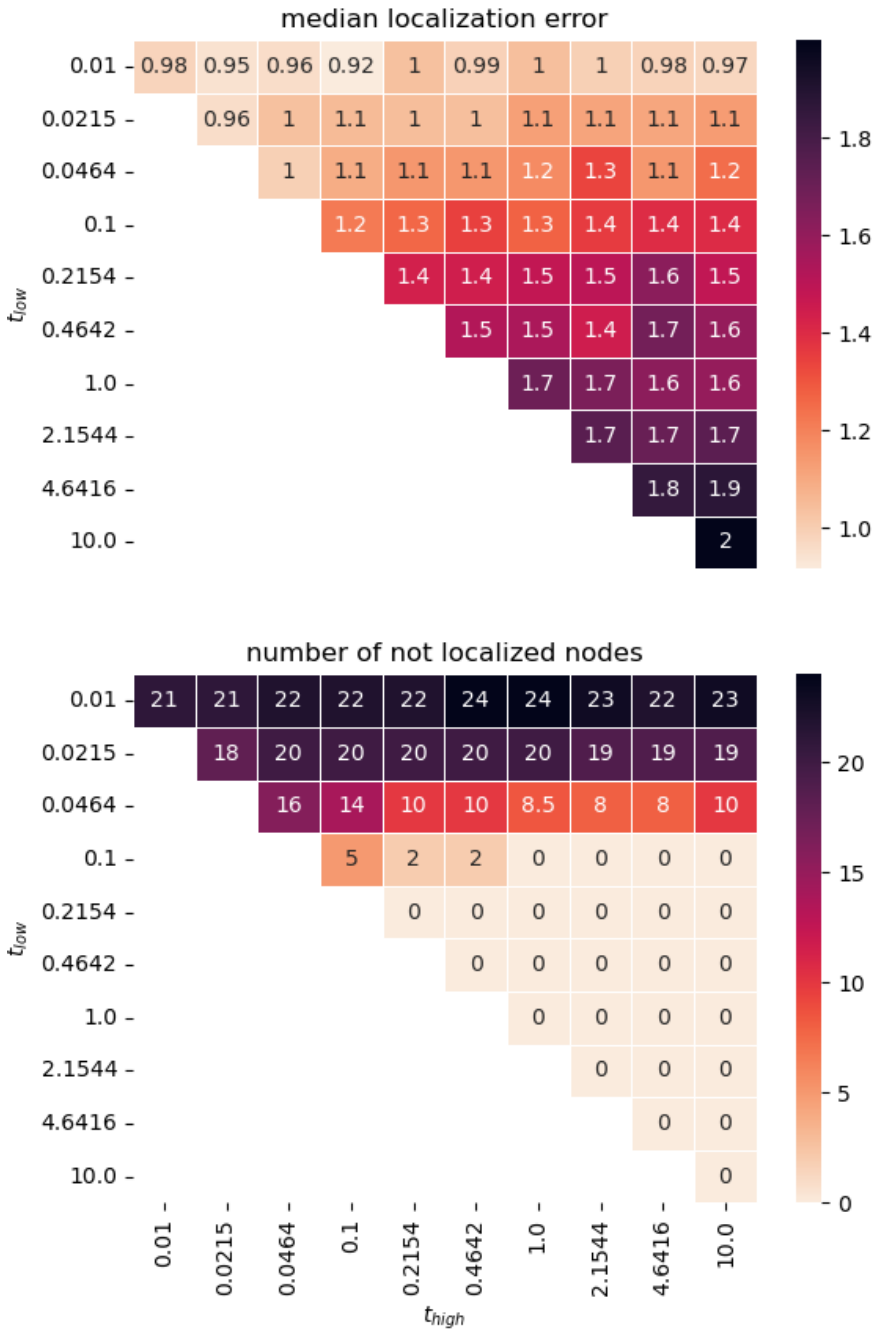

Figure 5.11: Median localization error in meters and number of not localized nodes using different thresholds for the **variance solver** metric as anchor decision algorithm (median over 31 runs)

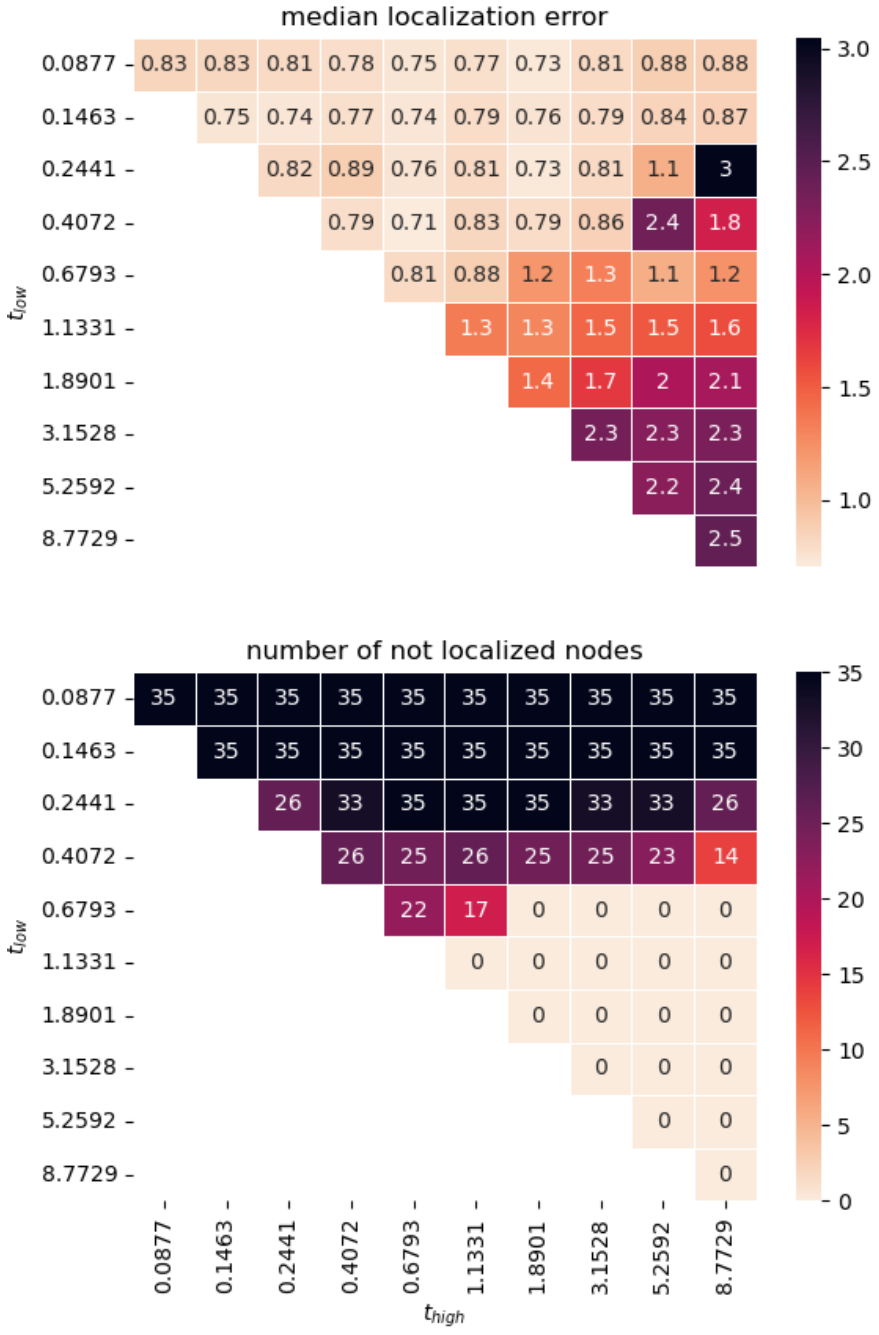

Figure 5.12: Median localization error in meters and number of not localized nodes using different thresholds for the **variance angle** metric as anchor decision algorithm (median over 31 runs)

5.5.3 Influence of the Maximum Number of Allocated Anchor Nodes

The proposed metrics are used as anchor selection algorithm by ranking available nodes around each node in that metric and then using the k best performing nodes as reference to enable the calculation of an position estimate.

In this experiment, the influence of this maximum anchor size (k) is evaluated. To do so, a multi hop scenario with eight original anchor nodes and 100 nodes is simulated. Again, as in previous experiments, the eight anchor nodes were randomly sampled onto an area of 10 by 10 meters, while the other nodes are randomly distributed over an area of 20 by 20 meters (as in figure 5.8). As anchor selection algorithms, the variance angle, variance solver and the hop count metric are used with different values for k. These configurations are tested with the variance solver metric as selection algorithm, as well as the variance angle metric. There, the previously determined values for t_{high} and t_{low} are used.

For the maximum number of anchors (k), values in the range from three to 20 are tested. At least three anchors are needed to compute a position in two dimensions. For higher values of k, fewer values are tested, to enable an exploration of the behavior without too much computational effort. Furthermore, it is expected, that for those values of k the behavior is going to converge since only a limited number of nodes is sending. When k is set too high, the anchor selection algorithms are forced to always take all available nodes, rendering its behavior equal to the "take all" approach (section 5.5.1).

All the resulting configurations were run 31 times for 2000 iterations with an allocation phase every 25 iterations. Each run was performed on the same initially randomly generated map with different randomly sampled solver starting positions.

Figure 5.13 shows the median position errors and not localized nodes for the different values of k over all combinations of the tested anchor selection and anchor decision functions at iteration 2000. It is immediately visible, that the maximum number of anchors has a huge impact on the overall performance. Solutions which manage to localize more nodes come with a higher median localization error, while solutions with lower median localization error tend to localize fewer nodes.

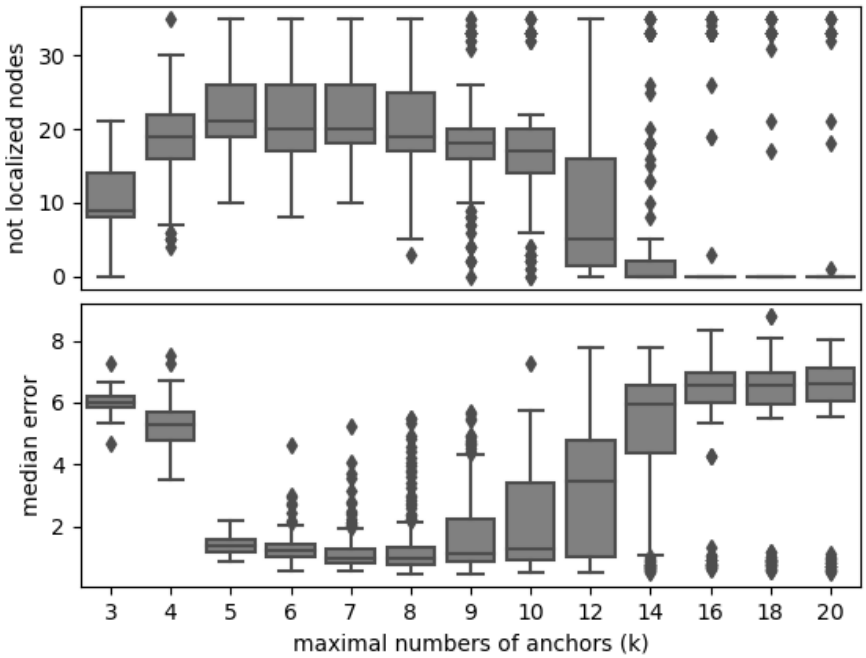

Figure 5.13: Distribution of median localization error in meters and number of not localized nodes at iteration 2000 for different maximum anchor values (k) for the selection functions

Values for k of 16 and greater change the overall behavior of the algorithms to mostly localize all nodes with a median localization error of around six to seven meters. This performance is similar to using all available anchor nodes as in the "all" approach in section 5.5.1. This backs up the previously stated assumption of high maximal anchor values (k) making the anchor selection algorithms always choosing all available anchors.

Using only three or four anchor nodes as reference results in similarly high median localization errors of five to six meters. A drastic decline in the localization error can be seen when increasing k to five or more. With $k = 9$, the median localization error visibly starts rising again, until it reaches it's maximum with $k >= 16$. Overall, similar behavior could be observed when looking at the specific configurations of algorithms rather than the combined plot of figure 5.13.

Using more anchors generally results in higher errors due to longer schedules and more influence of multi path distortions.

It is also likely that the increasing of the error beginning with $k = 9$ is related to the eight original anchors used. When choosing up to nine nodes whenever possible results in barely any nodes only relying on the original anchors anymore, thereby including worse performing nodes in their calculation. This results in higher errors of the firsts hops, which eventually carry on through the whole network. Furthermore, cycles are more likely to occur in the Position Dependency Graph. These effects are further increased with higher values for k.

Another reason for the increasing median error and the lower amount of not localized nodes for higher values of k lies in the anchor decision metrics. Both metrics behave differently with higher amount of nodes affecting the amount of sending nodes. In case of the angle metric, more nodes automatically result in a lower score, even if their distribution is considered equal (e.g. $\sigma^2(0, 0, 0, 0, 0, 2pi) = 5.4831$ while $\sigma^2(0, 0, 2pi) = 8.7729$). For the variance solver metric, with more anchor nodes selected, the influence of one anchor node's calculated position "running away" on the variance of the own calculated position is lower. This also leads to smaller metric values with more reference nodes. In both cases, more nodes are going to send when applying the same thresholds t_{low} and t_{high} further increasing the median position error due to the error model and in the meantime covering a bigger range and thereby localizing more nodes.

Interestingly, with $k = 3$, the amount of localized nodes is increased compared to higher maximum anchor values. This might be because with only three selected anchor nodes, the system is way more dynamic, resulting in nodes switching between sending and not sending more often. With this, more nodes would be localized, even if they get position updates only for a short time. It is also possible that the proposed anchor decision algorithms do not work as intended with this few nodes and nodes share their position even though they have a bad position estimate (similar to higher values for k).

For the tested parameters, lowest median localization errors were achieved with k set to seven and eight. Between the median error values of these two, no significant difference could be found when performing a Mann–Whitney U test (p=0.715). However, using eight anchors significantly reduced the amount of not localized nodes compared to $k = 7$ (Mann–Whitney U test, p=0.0024).

Given these results and the research question asking for the performance with a median localization error below one meter, setting the maximum number of anchors to eight is seen as optimal and is further on used.

5.5.4 Convergence Analysis

This experiment aims to explore the behavior of selected algorithms over the performed iterations. It is used to determine the length experiments need to be run to ensure meaningful results. Furthermore, the influence of how often an allocation phase is performed is tested.

As in previous experiments, eight setup anchor nodes are distributed onto an area the size of 10 times 10 meters, with 100 nodes distributed over 20 by 20 meters as in figure 5.8.

For this experiment, the variance solver and variance angle metric are used as anchor decision algorithms with the earlier determined thresholds ($t_{low} = t_{high} = 0.0215$ and $t_{low} = 0.6793, t_{high} = 1.1331$). As selection algorithm, the hop count approach has been used next to the two variance based metrics with k set to eight. Additionally, different settings of how often an allocation phase is performed are tested.

There, 5, 10, 25, 50, and 100 position updates per allocation (UPA) are considered. When performing only five position updates before the next allocation phase, the anchor decision and anchor selection algorithms are run more frequently. Setting the position updates per allocation phase to 100 however, results in a less frequent reorientation of the nodes probably making the system less responsive, but more precise.

To monitor long term behavior, the 31 runs per configuration were performed over 12000 iterations. For each run, the solver starting positions were randomly sampled.

Figure 5.14 shows the number of not localized nodes and the median localization error over the performed iterations with the variance solver metric as anchor decision, and the hop count metric as anchor selection algorithm. The plots show a steep decline of the two performance metrics over the first 200 iterations. After that, the median position error settles at around one meter, while the number of not localized nodes slowly continues to decrease.

Overall, the different amounts of position updates per allocation phase (UPA) do not appear to make a big difference considering the localized nodes. Setting UPA to five outperforms the other approaches at the first 1000 iterations. However, at around 2000 iterations, higher values for UPA surpass this approach in terms of localized nodes.

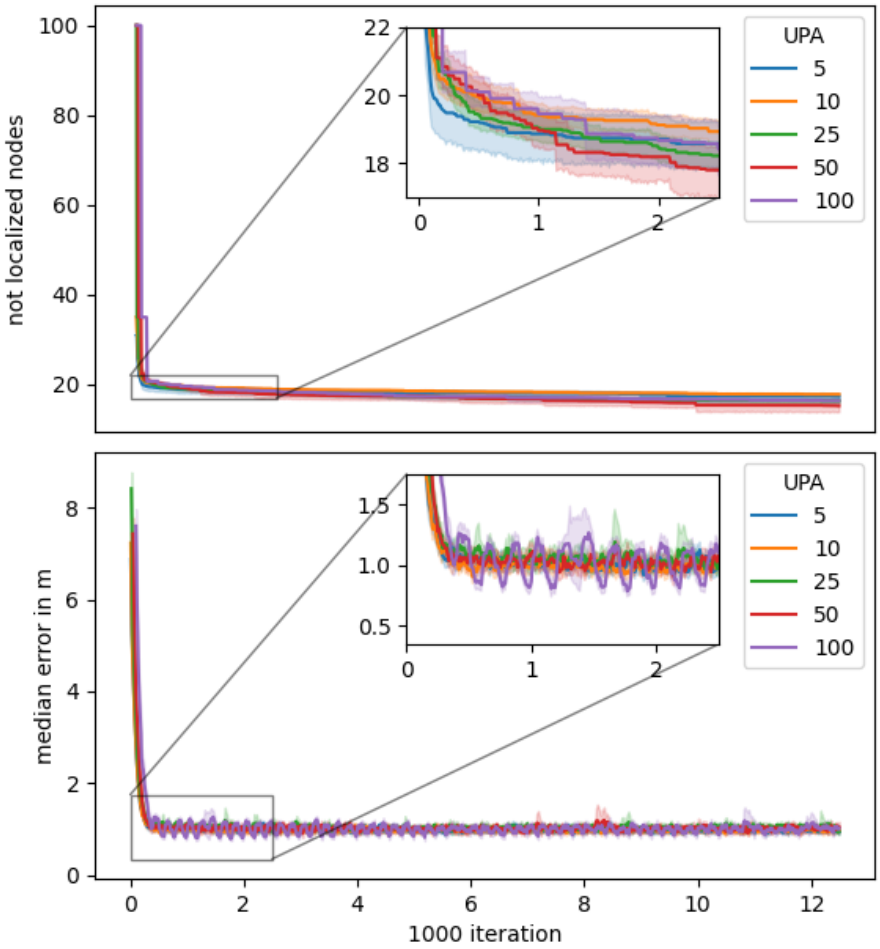

Figure 5.14: Convergence of the number of not localized nodes and the median localization error in meter for variance solver - hop count

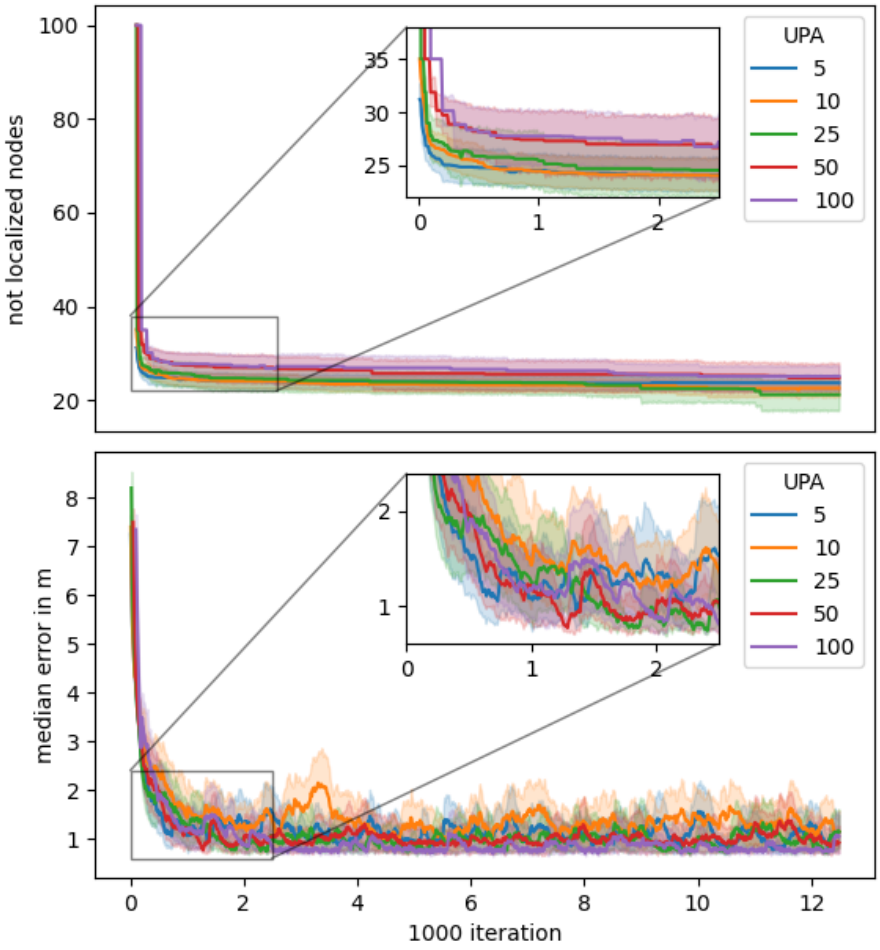

Figure 5.15: Convergence of the number of not localized nodes and the median localization error in meter for variance angle - hop count

Looking at the median localization error it stands out, that performing an allocation phase each 100 localization updates results in a swinging behavior roughly inline with the allocation phases. A similar but much less intense behavior can be observed with UPA set to 50. A possible explanation can be found when considering cycles in the Position Dependency Graph. Due to the low update rate of the chosen anchors, cycles would increase the error over the 100 (or 50 respectively) updates. After an allocation phase, new anchor nodes

would be used, first lowering the error, but then again increasing it given new cyclic dependencies. When performing the allocation phase more frequent, no such behavior could be observed. There the values of 5, 10 and 25 for UPA enable a similar performance regarding the median localization error.

Figure 5.15 visualizes the median localization error and the amount of not localized nodes for the combination of the variance angle metric as anchor decision algorithm, again with the hop count metric as anchor selection.

Regarding the number of not localized nodes, this approach shows a similar convergence behavior. However, it can be seen that performing the allocation phase less often by setting the position updates per allocation (UPA) to 50 or 100 results in less nodes localized. Again, the results with an UPA of five converge the quickest, while being matched by UPA 10 and 25 after the first 1000 iterations. Furthermore, the results vary a lot more compared to the approach using the variance solver metric as anchor decision algorithm.

This also can be observed for the median localization error. There the errors range from around 0.9 to two meters even when the algorithm is converged. In the first 2000 iterations, setting UPA to 25 appears to reach the lowest errors of below one meter, while UPA of five and 10 has the highest errors. When looking at iterations greater than 3000, running 100 localization updates per allocation phase seems to settle with the lowest median position errors, while 10 localization updates per allocation phase results in spikes in errors of two meters and more.

Similar convergence behavior has been observed for the other tested combinations of allocation selection and allocation decision functions and is not further considered in this section since the presented combinations are seen as more relevant in terms of the desired performance.

Overall, the approaches converged within the first 2000 iterations. Running experiments for more iterations could slightly increase the amount of localized nodes. However, the rather small expected performance gain would require a significantly higher computational effort, which does not justify the small increase in localized nodes. Other multi hop experiments are therefore run for 2000 iterations if possible.

Performing an allocation phase less often (UPA of 50 or 100) overall results in less desirable behavior compared to more frequent runs of the anchor decision

and anchor selection algorithms. Due to research pragmatic reasons, other experiments are conducted with 25 localization updates per allocation phase.

5.5.5 Performance on Different Scenarios

In this final experiment, the influence of different amounts of nodes and different sizes of the area the nodes are distributed over is examined.

As in previous experiments, eight anchor nodes are distributed over an area of 10 times 10 meters. There, the same positions as in experiment 5.5.1 are used. The different amounts of nodes were then distributed randomly on an area of 10, 15, 20 and 25 meters square. As in figure 5.8, the area on which the original anchors are sampled on is placed in one corner of the other. In case of the smallest tested size, the two areas are the same ($A_N = A_A$). Compared are the combination of the hop count metric with $k = 8$ as selection algorithm and the variance solver metric as anchor decision algorithm with $t_{low} = t_{high} = 0.0215$ against the single hop approach where only the eight original anchors are sending and tags always using all signals available. The combination of algorithms representing the multi hop approach has been chosen based on previous experiments. There it can be seen that this approach is able to locate more nodes than other algorithm pairs while still maintaining an median position error below one meter in the tested scenarios. Furthermore, this combination shows a more predictable behavior over the performed iterations, with less varying results compared to the variance angle metric as anchor decision algorithm.

Again, 31 runs over 2000 iterations were performed for each tested configuration. For each run, the starting positions of the solver are randomly sampled.

Figure 5.16 visualizes the median localization error and the amount of not localized nodes for the two tested approaches over different amounts of nodes and different area sizes the nodes are distributed over at iteration 2000. While the distribution of nodes over the different scenarios is not equal, the 31 runs of one configuration always use the same "map" of nodes.

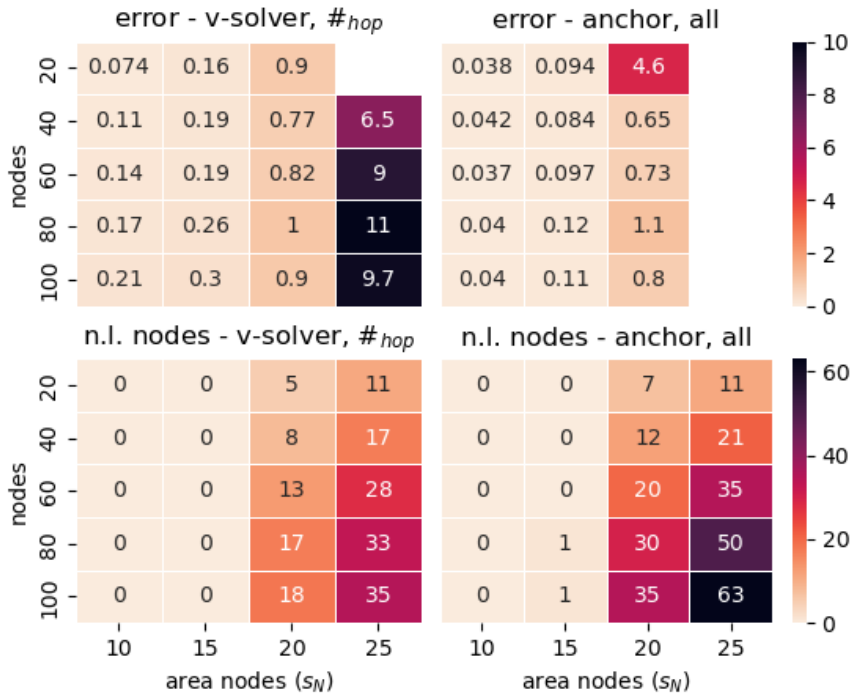

Figure 5.16: Scaling analysis of selected approaches showing median localization error in meter and not localized nodes over different areas and amount of nodes.

In figure 5.16, the upper two heat maps show the median error of both approaches. Again, the median localization error is the median over all nodes, also the ones which are not localized (excluding the original anchor nodes). Overall, it is noticeable that the errors increase with the size of the area the nodes are distributed over. For the largest tested area of 25 by 25 meters, the single hop approach is not able to locate more than half of the nodes. Therefore, no median localization error could be computed, represented by blank entries (figure 5.16 top right). The median localization errors of the multi hop approach (figure 5.16 top left) for the smaller areas does also seem to be impacted by the amount of nodes. This might be due to a faster convergence behavior when using smaller amounts of nodes. The two heat maps furthermore show that the median localization error of the smaller areas is drastically lower when using the single hop approach. This shows, that the proposed multi hop approach is not able to improve the localization performance in a scenario fully solvable with a single hop approach (research question Q2). However,

this changes when looking at lager areas. For the 20 by 20 meters sized area, the shown multi hop algorithms manage to roughly match the performance of the single hop approach for scenarios with 40 or more nodes. When looking at 20 nodes, it dramatically outperforms the single hop approach.

The lower two heat maps in figure 5.16 visualize the number of not localized nodes (n.l. nodes). It can be seen that the single hop approach manages to localize all nodes for the smallest tested size, while for the area of 15 by 15 meters, it locates every but one node for the scenarios with 80 and 100 nodes. However, for the larger area of 20 by 20 meters, between 30 and 35 percent of the nodes are not localized, which further increases to over 50 percent for the biggest area tested of 25 by 25 meters. Overall, similar percentages of not localized nodes can be seen across the different amounts of nodes tested.

The multi hop approach is able to significantly reduce the amount of not localized nodes. When considering scenarios with higher amounts of nodes, the multi hop approach reduces the amount of not localized nodes by around 45 percent. It does so while maintaining errors of one meter and lower for the 20 by 20 meter sized map. For the larger map however, the calculated positions have a very high median error of more than six meters. These high errors show the limit of the tested approach (research question Q4). However, choosing different parameters for the proposed algorithms may improve this scenario a lot, at the cost of higher localization errors on smaller areas.

It is noticeable that the multi hop approach is not able to localize more nodes compared to the single hop case when looking at the scenario with $s_N = 25$ and 20 nodes. This phenomenon can be explained with the low node density of this scenario. There it is not possible to find at least three reference nodes for each node due to the transmission range limitation of 10 meters.

Overall it can be seen, that the tested multi hop approach does not improve the localization performance in scenarios, where a single hop setup reaches all nodes. However, the multi hop approach is able to significantly improve the amount of localized nodes when increasing the area to the point, where a single hop scenario is not able to reach all nodes. For the tested scenario, it has been shown, that the tested approach manages to do so while maintaining median position errors of one meter and below on a map size of four times the area the original anchors are distributed on.

This experiment has also been run with more challenging positions of the original anchor nodes. These were mostly distributed in the lower left corner

of the 10 by 10 meter area, effectively sizing this area down to around six times six meters. Consequently, the single hop approach already struggled with the 15 by 15 meters map with roughly 15 to 20 percent of the nodes not localized, while the multi hop approach was able to localize all nodes for the scenario with higher amounts of placed nodes (>20) with median position errors of around 0.6 meters. Overall, the results are shifted towards smaller sizes of the map.

6 Conclusion

In this thesis, a passive TDOA localization approach using ultra wide band communication has been extended to be applicable in a multi hop scenario. Different methods for node allocation and position sharing have been proposed and evaluated in simulation.

To generate meaningful results, clock drifts of DWM1000 UWB modules have been recorded and used in two different ways as input for the simulation error model. It has been shown, that the normal distribution based approach is applicable.

After further determination of basic parameters, a performance overview of different methods to achieve multi hop localization is given in this work. There, it has become visible that sharing and using all available positions is not feasible. Other proposed approaches however manage to increase the amount of localized nodes in certain multi hop scenarios while maintaining similar median localization errors of below one meter compared to the single hop approach. Using a combination of the proposed hop count metric as anchor selection and variance solver metric as anchor decision algorithm showed about 45% less not localized nodes in a scenario where nodes were located on a four times lager area than the original anchor nodes (Q3). This has been achieved with a maximum of eight anchor nodes selected by the hop count metric and thresholds of $t_{low} = t_{high} = 0.0215$ for the variance solver metric.

Further analyzing the threshold parameters of the anchor decision algorithms showed a huge influence on the multi hop performance. Lower parameter values tend to decrease the median localization error but localize fewer nodes, while higher values enable the localization of more nodes but also generate larger errors. Thresholds of $t_{low} = 0.6793$ and $t_{high} = 1.8901$ for the variance angle and $t_{low} = 0.1$ and $t_{high} = 1.0$ for the variance solver metric were able to localize all nodes in the tested scenario with only slightly higher median localization errors of around 1.3 meters.

Additionally, the influence of the maximum amount of selected anchor nodes has been looked at. While using seven and eight anchors produced similarly low median localization errors, significantly more nodes could be localized when using eight anchor nodes. Choosing only three to four anchor nodes, as well as choosing more than nine, results in very high median localization errors for the tested scenarios.

Further experiments have demonstrated, that using the multi hop approach based on the variance solver decision and the hop count selection algorithm on an area, which is already properly covered with original anchors did not yield any benefit (Q2).

The disadvantage of multi hop localization is, that the position and synchronization errors of nodes are forwarded to other nodes, limiting the range which it can be used to extend localization. The trade-off when localizing more nodes or extending the area of the nodes is the higher overall localization error. This phenomenon also became apparent in the different results of this thesis. Due to the more complex timing and synchronization dependencies of the passive TDOA approach and with the conservative error model used, no meaningful results could be seen when extending the area to 25 by 25 meters with the tested settings (Q4).

Overall, this thesis shows working approaches to enable passive TDOA multi hop localization with possible application in systems with many nodes in GPS denied environments where the area scale is not yet completely known.

7 Future Work

The multi hop capable passive TDOA localization approach introduced in this thesis enables further research on this topic.

In this thesis, the proposed algorithms, variance solver and variance angle, have been shown to display very different behavior depending on the parameters chosen. Adjusting these parameters over time of nodes could drastically improve the overall performance of the system. Also, enabling tags to send a position request to trigger other anchor decision algorithms using this information is conceivable. Furthermore, different synchronization node allocation algorithms can be implemented and analyzed.

To make the approaches proposed in this thesis a more viable option for wireless sensor networks and certain swarm applications, an energy model and an adaption of the proposed algorithms with regard to power consumption and state of battery would be necessary. Additionally, the influence of moving nodes in the system can be analyzed. There it would be interesting to see how the system reacts to different amounts of moving nodes with different speeds and movement models.

In the end, an actual implementation of the proposed multi hop localization system on ultra wide band nodes with different scheduling approaches of the sending nodes would greatly contribute to research in this topic.

Bibliography

[1] 2d Lidar Sensor TIM510-9950000S01 | Mess- und Detektionslösungen | SICK. https://www.sick.com/de/de/mess-und-detektionsloesungen/2d-lidar-sensoren/tim5xx/tim510-9950000s01/p/p343047? Accessed: 2022-02-16.

[2] 3D localization technology | Optical tracking explained | 6DOF. https://www.ps-tech.com/optical-tracking-explained/. Accessed: 2022-02-26.

[3] Cobalt Robotics - Seamless robotic security services for all situations. https://www.cobaltrobotics.com/. Accessed: 2022-02-07.

[4] D-RTK 2 High Precision GNSS Mobile Station - DJI Store. https://store.dji.com/de/product/d-rtk-2-high-precision-gnss-mobile-station. Accessed: 2022-02-15.

[5] Datasheet laser distance sensor LDS-01. http://emanual.robotis.com/docs/en/platform/turtlebot3/appendix_lds_01/. Accessed: 2022-02-26.

[6] DWM1000 Qorvo | Mouser Germany. https://www.mouser.de/ProductDetail/Qorvo/DWM1000?qs=sGAEpiMZZMu3sxpa5v1qrrjDD2pjKdCcRPXEFIZh8rA%3D. Accessed: 2021-10-12.

[7] ECS-327MVATX 32.768 kHz Crystal Oszillator Datasheet Rev. 2021.

[8] GPS.gov: GPS Accuracy. https://www.gps.gov/systems/gps/performance/accuracy/. Accessed: 2022-02-15.

[9] The Implementation of Two Way Ranging with the DW1000 - APS013 Application Note Version 2.3.

[10] Indoor Positionsbestimmung - Basisinformationen von infsoft. https://www.infsoft.com/de/indoor-positionsbestimmung. Accessed: 2021-09-02.

[11] OptiTrack System Configurator. `http://www.optitrack.com/systems/index.html`. Accessed: 2022-02-16.

[12] Pallet Transport 1500 - Autonomous Pallet Conveyance. `https://fetchrobotics.com/pallettransport1500/`. Accessed: 2022-02-16.

[13] Two Way Ranging (TWR). `https://www.sewio.net/uwb-technology/two-way-ranging/`. Accessed: 2022-02-09.

[14] YDLIDAR X4 Abstandssensor. `https://www.conrad.de/de/p/ydlidar-x4-abstandssensor-1-st-reichweite-max-im-freifeld-10-m-l-x-b-x-h-101-7-x-71-3-x-58-4-mm-2522283.html`. Accessed: 2022-02-16.

[15] Accurate indoor positioning with Bluetooth beacons. `https://proximi.io/accurate-indoor-positioning-bluetooth-beacons/`, July 2017. Accessed: 2022-02-07.

[16] SPS Performance Analysis Report 2020 Q4. `https://www.nstb.tc.faa.gov/reports/2020_Q4_SPS_PAN_v2.0.pdf#page=38`, 2020. Accessed: 2022-02-07.

[17] Nuha A. S. Alwan and Zahir M. Hussain. *Gradient Descent Localization in Wireless Sensor Networks*. IntechOpen, October 2017. Publication Title: Wireless Sensor Networks - Insights and Innovations.

[18] Jonathan Beckhaus. Design of a test setup for three dimensional indoor positioning systems. 2019.

[19] Anja Bekkelien. Bluetooth indoor positioning, March 2012.

[20] Andrea Botta and Giuseppe Quaglia. Performance Analysis of Low-Cost Tracking System for Mobile Robots. *Machines*, 8(2):29, June 2020.

[21] Kenneth C Cheung, Stephen S Intille, and Kent Larson. An Inexpensive Bluetooth-Based Indoor Positioning Hack. In *Proceedings of UbiComp*, volume 6, 2006.

[22] Pablo Corbalán, Gian Pietro Picco, and Sameera Palipana. Chorus: Uwb concurrent transmissions for gps-like passive localization of countless targets. In *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 133–144. IEEE, 2019.

[23] F Serhan Daniş and Ali Taylan Cemgil. Model-Based Localization and Tracking Using Bluetooth Low-Energy Beacons. *Sensors*, 17(11):2484, 2017.

[24] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine*, 13(2):99–110, 2006.

[25] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, June 2014.

[26] Markus Hempel. Programming and Evaluation of an Ultra-Wideband Distance Measurement System for Mobile Robots.

[27] Benjamin Kempke, Pat Pannuto, and Prabal Dutta. PolyPoint: Guiding Indoor Quadrotors with Ultra-Wideband Localization. In *Proceedings of the 2nd International Workshop on Hot Topics in Wireless - HotWireless '15*, pages 16–20, Paris, France, 2015. ACM Press.

[28] Ajitesh Kumar. Convex optimization explained: Concepts & Examples. `https://vitalflux.com/convex-optimization-explained-concepts-examples/`, October 2021. Accessed: 2022-02-09.

[29] Vasileios Lakafosis and Manos M. Tentzeris. From single-to multihop: The status of wireless localization. *IEEE Microwave Magazine*, 10(7):34–41, December 2009. Conference Name: IEEE Microwave Magazine.

[30] Herbert Landau, Xiaoming Chen, Sören Klose, Rodrigo Leandro, and Ulrich Vollath. Trimble's Rtk And Dgps Solutions In Comparison With Precise Point Positioning. In Michael G. Sideris, editor, *Observing our Changing Earth*, International Association of Geodesy Symposia, pages 709–718, Berlin, Heidelberg, 2009. Springer.

[31] Decawave Ltd. *DWM1000 Data Sheet*. Decawave, 2016.

[32] Decawave Ltd. *DW1000 User Manual*. Decawave, 2017.

[33] Ivan A. Mantilla-Gaviria, Mauro Leonardi, Juan V. Balbastre-Tejedor, and Elías de los Reyes. On the application of singular value decomposition and Tikhonov regularization to ill-posed problems in hyperbolic passive location. *Mathematical and Computer Modelling*, 57(7):1999–2008, April 2013.

[34] Rainer Mautz and Sebastian Tilch. Survey of optical indoor positioning systems. In *2011 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–7, Guimaraes, Portugal, September 2011. IEEE.

[35] Pierre Merriaux, Yohan Dupuis, Rémi Boutteau, Pascal Vasseur, and Xavier Savatier. A study of vicon system positioning performance. *Sensors*, 17(7):1591, 2017.

[36] Brian O'Keefe. Finding Location with Time of Arrival and Time Difference of Arrival Techniques. 2017.

[37] Abdullah Raouf. Minimize Frequency Drift In Crystals. `https://www.electronicdesign.com/technologies/analog/article/21798809/minimize-frequency-drift-in-crystals`, December 2013. Accessed: 2022-02-10.

[38] Patrick Schober, Christa Boer, and Lothar A Schwarte. Correlation coefficients: appropriate use and interpretation. *Anesthesia & Analgesia*, 126(5):1763–1768, 2018.

[39] Juri Sidorenko, Volker Schatz, Norbert Scherer-Negenborn, Michael Arens, and Urs Hugentobler. Decawave uwb clock drift correction and power self-calibration. *Sensors*, 2019.

[40] Christoph Steup, Jonathan Beckhaus, and Sanaz Mostaghim. A single-copter uwb-ranging-based localization system extendable to a swarm of drones. *Drones*, 2021.

[41] Janis Tiemann, Fabian Eckermann, and Christian Wietfeld. Multi-user interference and wireless clock synchronization in TDOA-based UWB localization. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–6, 2016.

[42] Janis Tiemann, Florian Schweikowski, and Christian Wietfeld. Design of an UWB indoor-positioning system for UAV navigation in GNSS-denied environments. In *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7, 2015.

[43] PYP Tsang, Chun-Ho Wu, WH Ip, G Ho, and Y Tse. A bluetooth-based indoor positioning system: a simple and rapid approach. *Annual Journal IIE (HK)*, 35(2014):11–26, 2015.

[44] Frank van Diggelen and Per Enge. The World's first GPS MOOC and Worldwide Laboratory using Smartphones. pages 361–369, September 2015. ISSN: 2331-5954.

[45] Mengda Wang, Bing Xue, Wei Wang, and Junjie Yang. The design of multi-user indoor UWB localization system. In *2017 2nd International Conference on Frontiers of Sensors Technologies (ICFST)*, pages 322–326, 2017.

[46] Futa Watanabe. Wireless sensor network localization using aoa measurements with two-step error variance-weighted least squares. *IEEE Access*, 9:10820–10828, 2021.

[47] C. Yang and H. Shao. WiFi-based indoor positioning. *IEEE Communications Magazine*, 53(3):150–157, March 2015.

Declaration of Authorship

I hereby declare that this thesis was created by me and me alone using only the stated sources and tools.

Jonathan Beckhaus Magdeburg, 28.02.2022