

# Robust Reconstruction of Sensor Swarms Floating through Enclosed Environments

Erik H. A. Duisterwinkel<sup>1,2\*</sup>, Gijs Dubbelman<sup>1</sup>, Libertario Demi<sup>1</sup>, Elena Talnishnikh<sup>2</sup>, Heinrich J. Wörtche<sup>1,2</sup>, Jan W. M. Bergmans<sup>1</sup>

<sup>1</sup>Eindhoven University of Technology, Eindhoven, The Netherlands

<sup>2</sup>INCAS3, Assen, The Netherlands

Email: \*mail@erikduisterwinkel.nl

**How to cite this paper:** Duisterwinkel, E.H.A., Dubbelman, G., Demi, L., Talnishnikh, E., Wörtche, H.J. and Bergmans, J.W.M. (2018) Robust Reconstruction of Sensor Swarms Floating through Enclosed Environments. *Wireless Sensor Network*, 10, 1-39.

<https://doi.org/10.4236/wsn.2018.101001>

**Received:** November 20, 2017

**Accepted:** January 28, 2018

**Published:** January 31, 2018

Copyright © 2018 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

A novel type of application for the exploration of enclosed or otherwise difficult to access environments requires large quantities of miniaturized sensor nodes to perform measurements while they traverse the environment in a “go with the flow” approach. Examples of these are the exploration of underground cavities and the inspection of industrial pipelines or mixing tanks, all of which have in common that the environments are difficult to access and do not allow position determination using e.g. GPS or similar techniques. The sensor nodes need to be scaled down towards the millimetre range in order to physically fit through the narrowest of parts in the environments and should measure distances between each other in order to enable the reconstruction of their positions relative to each other in offline analysis. Reaching those levels of miniaturization and enabling reconstruction functionality requires: 1) novel reconstruction algorithms that can deal with the specific measurement limitations and imperfections of millimetre-sized nodes, and 2) improved understanding of the relation between the highly constraint hardware design space of the sensor nodes and the reconstruction algorithms. To this end, this work provides a novel and highly robust sensor swarm reconstruction algorithm and studies the effect of hardware design trade-offs on its performance. Our findings based on extensive simulations, which push the reconstruction algorithm to its breaking point, provide important guidelines for the future development of millimetre-sized sensor nodes.

## Keywords

Sensor Swarm, Offline Reconstruction, Guided-RANSAC, Non-Unique Identifiers

---

## 1. Introduction

Automatic and detailed reconstruction of environments and objects from sensory

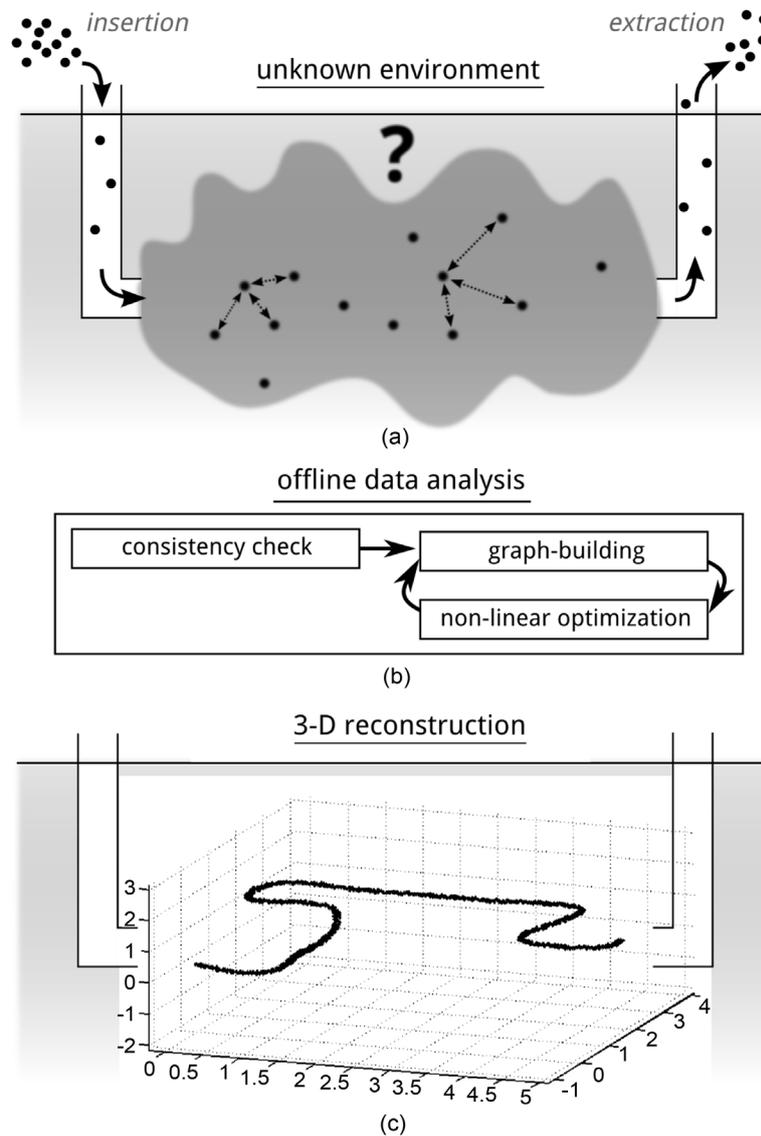
data is one of the success stories of modern signal processing. Still there are environments for which detailed reconstruction is not possible with current remote or *in situ* sensing technologies. In this work, we focus on such environments and examples are: the interior of deep underground formations like (oil-) reservoirs, mines and geothermal sources, and industrial infrastructure like pipelines, mixing tanks and reactors. These environments are filled with a (semi-) liquid medium and typically are globally large but locally small (e.g. in diameter). Furthermore, they are difficult-to-access for *in situ* measurements and unsuitable for remote sensing methods. In this paper we consider a go with the flow approach that follows a sensor swarm paradigm, in which large quantities (e.g. hundreds or thousands) of highly miniaturized and redundant sensor nodes are gradually inserted into the environment [1] [2]. Carried by the flow of the medium, the swarm fills the environment and by reconstructing the node positions, and therewith the shape of the sensor swarm, the geometric structure and size of the environment can be inferred. Reconstruction of the sensor swarm is performed after nodes are extracted from the environment and their data read-out. This process is illustrated in **Figure 1**.

The individual sensor node positions relative to each other can be reconstructed from sparse inter-node distance measurements [3] [4]. These distance measurements can be obtained using a ranging protocol as in e.g. [5]. As illustrated in **Figure 2**, reconstructing the positions of the nodes can be seen as a specific kind of graph problem,  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where the vertices  $\mathcal{V}$  are the sensor node positions and the edges  $\mathcal{E}$  are the distances between them. Typically such a problem can be solved using non-linear optimization algorithms [28], however, after obtaining the measured distances (*i.e.* the edges  $\mathcal{E}$  in graph  $\mathcal{G}$ ), the challenge is to obtain a robust initial estimate of the node positions in order for non-linear graph optimization algorithms to succeed. In order to robustly obtain such an initial estimate, our offline<sup>1</sup> reconstruction algorithm uses a Random Sampling Consensus (RANSAC) method applied to general lateration techniques [6] [7]. To the best of our knowledge, the novel reconstruction algorithm presented in this work, is one of the most robust algorithms that is able to perform 3-D reconstruction of large sensor swarms. Our applications require such a robust reconstruction algorithm since the distance measurements are performed under very adverse sensing conditions, as explained in Section 2. The details of our novel method are provided in Section 3 and Section 4.

We use this reconstruction algorithm to study trade-offs in the hardware design space, as many hardware challenges still need to be addressed before large swarms of millimetre-sized functional sensor nodes can be developed and deployed effectively. Trade-offs must be made in the hardware design with respect to sensing capabilities and size. Improved capabilities, e.g. larger

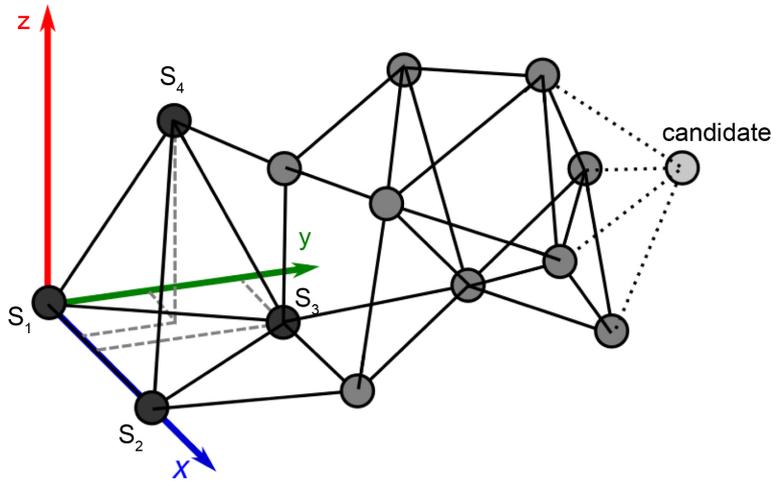
---

<sup>1</sup>In this paper we use the term *online* to indicate the period where nodes are in the process of performing measurements and while going through the environment, and the term *offline* for after the nodes are retrieved and data is extracted from the nodes.



**Figure 1.** A swarm of sensor nodes is inserted into the environment of interest; the sensor nodes traverse the environment using the flow of the medium. Once the sensor nodes are distributed in the environment, distance measurements between neighbouring sensor nodes are performed and stored locally within the nodes. Finally, the sensor nodes and their data are retrieved. All data is analysed offline to obtain the 3-D reconstruction of the sensor swarm from the inter-node distance measurements.

communication radius or better signal-to-noise ratios, will improve reconstruction performance, but make cost-effective miniaturization of nodes more challenging. Therefore, in Section 5, we use extensive simulations with increasingly more limitations on the hardware capabilities to study their effect on the reconstruction performance. These limitations include increased noise in the measurements and not being able to uniquely identify other nodes. The assessment of the performance of the reconstruction depends on which information is favoured from the swarm and different performance metrics are



**Figure 2.** Reconstructing nodes within a swarm can be seen as a graph problem,  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , with the node positions as vertices  $\mathcal{V}$  (circles) and the distances between nodes the edges  $\mathcal{E}$  (lines). The initial four node positions  $\{s_1, s_2, s_3, s_4\}$  are chosen to define the coordinate system to solve the global reflection and rotation ambiguity. Using general lateration techniques, additional nodes, like the candidate node in the figure, can be added when distances to at least four already reconstructed nodes are known. For clarity of the figure, not all (required) edges are drawn.

introduced.

The simulations in this work do not guarantee that robust reconstruction will work in reality, but as also discussed in Section 6, they do show that advances in offline reconstruction algorithms, can, to a large extent, compensate for severely limited hardware capabilities. Understanding the trade-offs between hardware design and the reconstruction performance can guide the hardware design of yet-to-be-developed highly miniaturized sensor nodes of future applications [2].

Several applications and research studies already exist that use a sensor swarm approaches where sensor nodes move with a flow through specific environments. In [8], 13 cm-sized sensor nodes were developed and deployed in the ocean to flow with the ocean currents and allows to study e.g. plankton behaviour. These nodes can actively change their density to adjust their depth while collecting a variety of sensor data. The nodes receive ultrasound ping signals from nearby buoys (at kilometre distance) such that afterwards their positions can be reconstructed using general lateration. The researchers in [9] developed millimetre-sized sensor nodes that can be injected into living fish to study how e.g. dams and ocean energy devices affect their behaviour. These sensor nodes measure temperature and emit a unique identifier using ultrasound such that they can be remotely tracked by an array of external hydrophones on the shoreline (within hundreds of metres distance). Both of the mentioned studies are characterized by the fact that external localization hardware (buoys or hydrophones with a known location) are used to reconstruct the node positions.

In our earlier work, targeting enclosed and difficult to access environments,

we developed 4 cm-sized sensor nodes that measure the node's inertial data (*i.e.* linear acceleration and rotation) and measure the local magnetic field. These nodes with variable density have been used *e.g.* to explore and inspect an underground pipeline of 15 centimetre in diameter and 260 metre in length [10] and to study the dynamics of multi-phase fluids in a mixing tank of several cubic metres [11]. Another of our attempts to work towards the mentioned future applications is described in [1]. It describes a field test that uses dummy (non-functional) sensor nodes to explore a deep underground sandstone oil-reservoir. This field test revealed that in order for the sensor nodes to pass through the 300 metre sandstone oil-reservoir, the size of sensor nodes cannot exceed 9 mm as the local dimensions are not larger than that.

The mentioned studies show that a go with the flow sensor approach can yield valuable insights into yet undiscovered environments and dynamics. They also show that some applications have clear constraints on *e.g.* the allowed maximum size of the used sensor nodes in order to pass through the environment. The objective of this paper is to see how far we can limit the capabilities of sensor nodes and how far they can be scaled down while still getting good reconstruction performance. In the following section (Section 2) we discuss the implications of scaling down sensor nodes to the millimetre size and operating them in enclosed and difficult to access environments.

Even though the used constraints in this paper represent a very challenging application scenario and might seem excessive in light of current applications, we believe they are valid constraints for future miniaturized nodes [2]. We think that providing a solution to the swarm reconstruction problem under these adverse sensing conditions can also be beneficial for current applications and problems. To summarize our contributions: 1) a novel and robust reconstruction algorithms is developed that deals with severe sensing conditions and imperfections, even including non-unique identification of distance measurements; 2) sensor node system design principles are derived from extensive simulations by sweeping input parameters until the reconstruction algorithm fails to reconstruct the sensor swarm; 3) different methods on how to assess the performance of the swarm reconstruction are provided, depending on which information is favoured from the swarm.

## 2. The Implications of Application Constraints

This section discusses existing work and shows that the applications which we target pose specific challenges that have not been addressed earlier in a holistic manner. Reconstructing the position of nodes in (wireless) sensor and robot networks has been subject of study for an extensive period of time [3] [4]. Obtaining simultaneously the location of nodes as well as obtaining the map of the yet-unknown environment, is often referred to as Simultaneous Localization and Mapping (SLAM) [12]. Our research also falls under this denominator but has to deal with several distinctive constraints relative to other work. These

constraints will be discussed in this section and can be summarized as:

- 1) no data is communicated between nodes (other than needed for distance measurements), measurements are stored in memory for offline analysis;
- 2) sparse connectivity in a large swarm;
- 3) not depending on external communication to e.g. fixed beacons;
- 4) using solely distance measurements, *i.e.* no direction information or additional sensor information, like bearing, odometry or inertia;
- 5) nodes are not identified uniquely.

Constraints 1, 3, and 4 are *hard* constraints and depending on the choice of applications, these either exist or not and make the reconstruction problem harder. Constraint 2 and 5 are *soft* constraints and can vary in severity, depending on e.g. parameter choices that can be tweaked. For example, considering constraint 2: the connectivity between nodes depends on the transmission power, the distance between nodes and signal-to-noise ratio at the receiving side. Considering constraint 5: the identifiability of nodes depends on the number of identifiers chosen relative to the number of nodes used. Increasing the severity of these soft constraints makes the reconstruction problem gradually harder. Therefore, in our study, the parameters influencing connectivity and identifiability are swept over a large enough range in order to study the consequences of this on the reconstruction performance and therewith also to find the limits of current reconstruction algorithms.

Although each separate constraint has been considered earlier; this work is, to the best of our knowledge, the first to research reconstruction of sensor swarms when considering all mentioned constraints together and under realistic conditions.

## 2.1. Power Consumption

Probably the most significant and overarching constraint of highly miniaturized sensor nodes is their power limitation. Due to volume and weight constraints in the small sensor nodes, battery capacity will be extremely limited and this influences all other design choices related to sensing, computation, and communication. In fact, we envision levels of miniaturization in which nodes have just enough energy to perform only a single inter-node distance measurement.

Furthermore, communication requires relatively large amounts of energy and is therefore only used to realize the inter-node distance measurements, but for example not for online distributed processing of the measurements (hence constraint 1). Instead of distributing the measured data or sending it to a central *sink node*, we use an approach in which the measured data is stored locally in the sensor nodes, and afterwards retrieved and processed centrally when power consumption is no longer a limitation.

Communication transmission power and the signal to noise ratio of receiving electronics are limited, reducing the communication range between nodes.

Therefore, the inter-node connectivity between sensor nodes, by means of inter-node distance measurements, will be sparse (hence constraint 2) and dependent on the energy budget.

To the best of our knowledge, there is no other research that aims to perform swarm reconstruction under such extreme low-power conditions.

## 2.2. Beacons and Landmarks

A distinction that can be made in SLAM related research is whether or not *beacons* are being used, that is, fixed points in space that serve as reference points where individual nodes can relate to. In our applications, we can not rely on the presence of beacons, as opposed to e.g. [13] [14] and the earlier mentioned work in [8] [9]. In our applications the environments are generally difficult-to-access and block long-range communication signals (hence constraint 3).

In applications with a static environment, the environment could potentially be used as a continuum of beacons, or in this case better described as *landmarks*. Opposed to the research mentioned in e.g. [15] where high-end cameras and/or laser range-finders are used, measuring clear fixed landmarks is infeasible as such detection devices have a high power consumption and need continuous measuring; it is something that doesn't scale well to miniaturized nodes. Furthermore, detection of the environment using such sensors as e.g. in [15] [16] would require clear line-of-sight in a transparent medium, which cannot be guaranteed in our applications.

Efforts to map node positions and the environments based on acoustic reflections can be found e.g. in [17] [18]. This, however, requires constraints on e.g. the node positions or the environment shape (e.g. straight walls), both of which cannot be guaranteed in our applications.

Given the aforementioned, the nodes are forced to cooperate with each other in order to obtain information on their positions, this also called cooperative localization [3].

## 2.3. Distance Measurements

A common method for cooperative localization is to reconstruct node positions based on distance measurements between the nodes. Often such methods are amended with e.g. inertial, bearing, or odometric information from the nodes, like in e.g. [19] [20]. In our research however, we explicitly rely only on inter-node distance measurements, as adding such devices or sensors would add significantly to the size and energy budget (hence constraint 4). For example, an inertial sensor would need to be active continuously to track the absolute position and orientation. Furthermore, current miniaturized inertial sensors are not accurate enough for absolute position determination for longer than e.g. one second. Alternatively, the absolute orientation of the node could be estimated using e.g. small and low-budget sensors that measure the Earth magnetic field

and the Earth gravitational field. But unfortunately, in our applications, no guarantees can be given on the external fields as the unknown environment can block or distort these.

Although one can argue that, in the future, advances in sensors will lift some of the mentioned limitations, we show that with only inter-node distance measurements, the reconstruction of the sensor swarm is possible.

## 2.4. Communication and Identification

Nodes with (sub-)centimetre dimensions only fit antennas that effectively produce EM radiation at frequencies that have a large absorption in the liquid media. The (liquid) media can have a high salinity and are also not guaranteed to be clear enough for the use of (visible) light communication. This effectively prevents EM communication for distances larger than e.g. a centimetre.

Ultrasound transducers at these scales do provide larger communication ranges within the boundaries of the environments, but yield other challenges for stable and fast communication between dynamic nodes in enclosed environments [21]. The information that can be communicated in a unit of time is therefore very limited, typically in the order of kilobytes per second or less.

Furthermore, the relatively slow speed of sound, in combination with the relatively small bandwidth, reflective environments and the fact that nodes are non-static, leads to highly time-varying communication channels [5] [22]. Distance measurements based on round-trip time of flight need to be completed within just a tiny fraction of a second as otherwise the nodes may have moved significantly relative to each other, invalidating the measured distance.

The low data rate and the need to perform distance measurements in a short period of time creates the incentive to reduce the content of the messages used for distance measurements (*i.e.* reducing the number of bits in the message). For example, in the ranging protocol as studied in [5] [22], at a data rate of 40 kbit/sec, the transmission of every additional bit for the identifiers to indicate the sender and receiver of the message adds 3 percent to the measurement noise caused by the movement of the nodes. This effect becomes larger when considering lower data rates. Furthermore, increasing the message length increases the probability of overlapping messages due to (time-varying) multi-path, thereby making it harder to correctly decode the messages.

In applications where many sensor nodes are used, ideally each node has a unique identifier by which it can unambiguously communicate with other nodes. However, for the envisioned application cases mentioned in this paper, it is worthwhile to explore the possibility of reducing the number of identifying bits as they make up the majority of the ultrasound messages (hence constraint 5). Reducing the number of identifying bits, and thus only having non-unique identification makes communication ambiguous. The inter-node distance measurements then also become ambiguous, but can be resolved using the robust reconstruction algorithm presented in this paper. Non-unique identification

of distance measurements makes most existing reconstruction efforts not applicable as they depend on well-defined, non-ambiguous identifications [23] [24] [25]. Earlier efforts to first resolve these ambiguities before the reconstruction of node positions can be found in [26], but does not include actual reconstruction results.

Our reconstruction algorithm is, to the best our knowledge, the first that can successfully cope with identification ambiguities and with all the other mentioned hardware constraints.

### 3. Problem Description

In this section we discuss the specific problem formulation and introduce our notation. All symbols used in this paper are summarized in **Table 1**. In this paper, whenever we use the bar-indicator, e.g.  $\bar{d}$ , it is to stress the true value of the parameter, to differentiate it from an estimated or measured value, e.g.  $d$ .

**Table 1.** Legend of symbols used throughout the paper.

Symbol	Description
$N$	Number of nodes in swarm
$n_{\text{CID}}$	Number of communication identifiers in use
$n_{\text{comm}}$	Number of nodes within communication radius
$r_{\text{comm}}$	Communication radius of nodes
$\mathcal{G}(\mathcal{V}, \mathcal{E})$	Graph with reconstructed nodes indicating positions as vertex $\mathcal{V}$ and distance between nodes as edges $\mathcal{E}$
$i, j$	Unique (hardware) identification number (UID) of nodes $i$ and $j$ , other letters are also used to indicate other nodes. An additional bar in the indication stresses these are the true values (opposed to measured or hypothesised)
$\text{CID}(i)$	Communication identification number (CID) used by node $i$
$d_{i,j}$	Distance between node $i$ and node $j$ . Additional bars in the indication stress that these are true values (opposed to measured or hypothesised).
$s_i$	Position of node $i$
$\mathcal{C}$	Consistency function used for associating measured distances from two nodes into a mutual connection
$\epsilon_r$	Threshold value for maximum allowed distance deviation in consistency condition $\mathcal{C}$
$h_{i,j}$	Set of hypothesised distances between node $i$ and $j$
$\mathbb{A}_i$	Number of additional identity ambiguities
$\mathbb{A}_D$	Number of additional distance ambiguities
$a \in \mathcal{A}, b \in \mathcal{B}, c \in \mathcal{C}$	Individual nodes (small letter) in a set of nodes (capital) selected based on some criteria in reconstruction algorithm
$\mathcal{Q}, \mathcal{Q}(b)$	Cliques in already reconstructed graph
$X, Y, Z$	Recall percentages of nodes that have been reconstructed; $Y$ and $Z$ have an additional condition on the accuracy of the reconstruction
$\mathbb{E}_{\text{abs}}, \mathbb{E}_{\text{glob}}, \mathbb{E}_{\text{loc}}$	Error metrics of a reconstruction indicating the mean absolute squared error of the reconstructed positions
$\mathbb{E}_{\text{rel}}$	Error metric of a reconstruction indicating the relative error of the reconstructed distances
$\nu$	Different noise parameters of distance measurements, expressed in percentages. Index indicates type of noise.

As mentioned in the introduction, the reconstruction of the node positions can be seen as a graph-optimization problem where  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , in which the vertices  $\mathcal{V}$  are the sensor node positions and the edges  $\mathcal{E}$  are the distances between them. This is also visualized in **Figure 2**. However, the edges of this graph can only be based on the *measured distances* that are noisy representations of the *actual distances*. In order to estimate an accurate graph of the nodes' positions we attempt to reduce the least squares error:

$$\arg \min_{\mathcal{V} \in \mathbb{R}^3} \sum_{\mathcal{E}} (\|\mathcal{V}_i - \mathcal{V}_j\| - \mathcal{E}_{i,j})^2 \quad (1)$$

Solving these kind of problems is often performed using non-linear optimization methods like e.g. a Levenberg-Marquardt algorithm or a Gauss-Newton algorithm. As with most non-linear optimization problems the key is to provide a proper initial estimate, in this case an initial estimate of the node positions. This initial estimate is then further refined using non-linear optimization. Obtaining the initial estimate for the node positions is performed by our novel robust reconstruction algorithm that is discussed in Section 4.

This section will first discuss the method on how to measure distances between the nodes (*i.e.* the edges  $\mathcal{E}$ ) and the difficulties herein. In Section 3.1 we discuss how the distance measurements are performed. Section 3.2 discusses the different types of measurement noise to be expected in these measurements and the issues related to identification are discussed in Section 3.3. The consequences of these on the reconstruction algorithm are discussed in Section 3.4.

### 3.1. Distance Measurement Protocol

Following the operational procedure as depicted in **Figure 1**, once the nodes are well distributed within the environment of interest, the nodes perform a single distance measurement to neighbouring nodes. This distribution can be achieved by e.g. gradually inserting new nodes into the environment until a steady outflow of nodes is reached. How the actual measurements are performed is defined by the *ranging protocol* that is used. Many of these protocols already exist and the one that is designed specifically for our applications is described in [5] [22].

Omnidirectional emission and reception is considered (as opposed to directional), because there is no a priori or online knowledge of the positions of neighbouring nodes. Whether or not messages are received by neighbouring nodes is dependent on the transmitted power and, among other factors, the received signal-to-noise ratio.

For convenience, but without loss of generality, in this paper we assume that distances can and are measured between all neighbouring nodes that are in line-of-sight and are within a fixed and known communication radius,  $r_{\text{comm}}$ .

Details of these protocols, as e.g. discussed in [22], are outside the scope of this paper, however, it is important to know that the messages that are

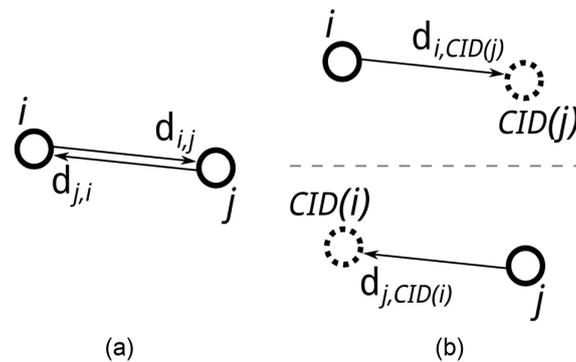
transmitted and received between the nodes for measuring the distance between them need to contain identifiers. The identifiers are used to identify the sender of the message, and depending on the protocol also to verify whether the message is actually addressed to the receiving nodes. These identifiers make up the largest part (in bits) of the message.

The result of the ranging protocol is a set of measurements where every node  $i$  measures the distance  $d_{i,j}$  to every neighbouring node  $j$ , and node  $j$  having measured the distance  $d_{j,i}$  to node  $i$ . This is visualized in **Figure 3(a)**. The measurements are individually stored in the nodes' memory.

### 3.2. Distance Uncertainty

The measured distance between nodes will be affected by different imperfections in the system. As the dynamics of the nodes and the environment they are in are unknown, a range of possible imperfections should be accounted for as we do not know the effect on the reconstruction result. To simulate the measured distances, in this paper we consider a single snapshot of the actual distances and introduce all types of measurement noise to account for the existing imperfections. The types of noise in the distance measurements that we are modelling can be divided in inlier-type noise and outlier-type noise. The inlier-type measurement noise being:

- additive Gaussian noise  $\mathcal{N}_a$ ; to account for e.g. offset in ranging timer or variable delay in electronics, and also to account for the errors due to movement of the nodes while performing the ranging measurements,
- multiplicative Gaussian noise  $\mathcal{N}_m$ ; to account for e.g. different clock



**Figure 3.** Mutual connections from distance measurements (indicated with arrows) with unique and non-unique identification. Distance measurements  $d_{i,j}$  as measured by node  $i$  and distance measurement  $d_{j,i}$  as measured by node  $j$  using unique identification can unambiguously be associated with each other. Distance measurements  $d_{i,CID(j)}$  as measured by node  $i$  and distance measurement  $d_{j,CID(i)}$  as measured by node  $j$  using non-unique identification can not be unambiguously associated with each other as more nodes use the same CID.

frequencies or inhomogeneous medium,

And the outlier-type measurement noise being:

- identification noise; to account for erroneous detection of identifiers,
- outlier noise; to account for burst-like/spiky/intermittent measurement noise.

The environment also gives rise to additional outlier-types of noise that can influence the reconstruction:

- obstruction of signal paths,
- reflection of signals,
- loss of nodes or inability to retrieve them.

How exactly these are simulated is described later, in Section 5.2.

### 3.3. Identification Uncertainty

Due to the constraints described in Section 2, the identifiers used in the ranging messages are reduced to non-unique identifiers, similar to the research in [5]. Each of the  $N$  nodes has a unique (hardware) *identification number*, UID, and is assigned a non-unique *communication identifier*, cid, that is used in the ranging message to identify the sender and addressee of the message. The number of available cids is indicated with  $n_{\text{CID}}$  and can be set as desired,  $n_{\text{CID}} \leq N$ . The available cids are distributed uniformly among the nodes.

Now, instead of every node  $i$  measuring distance  $d_{i,j}$  to node  $j$  and vice versa, node  $i$  is measuring distance  $d_{i,\text{CID}(j)}$  to a node that has a communication identifier  $\text{CID}(j)$ . And vice versa, node  $j$  measures distance  $d_{j,\text{CID}(i)}$ . This is illustrated in **Figure 3(b)**. The mapping from UID to CID is known and well defined, however, the mapping from CID to UID is ambiguous as more nodes share the same cid.

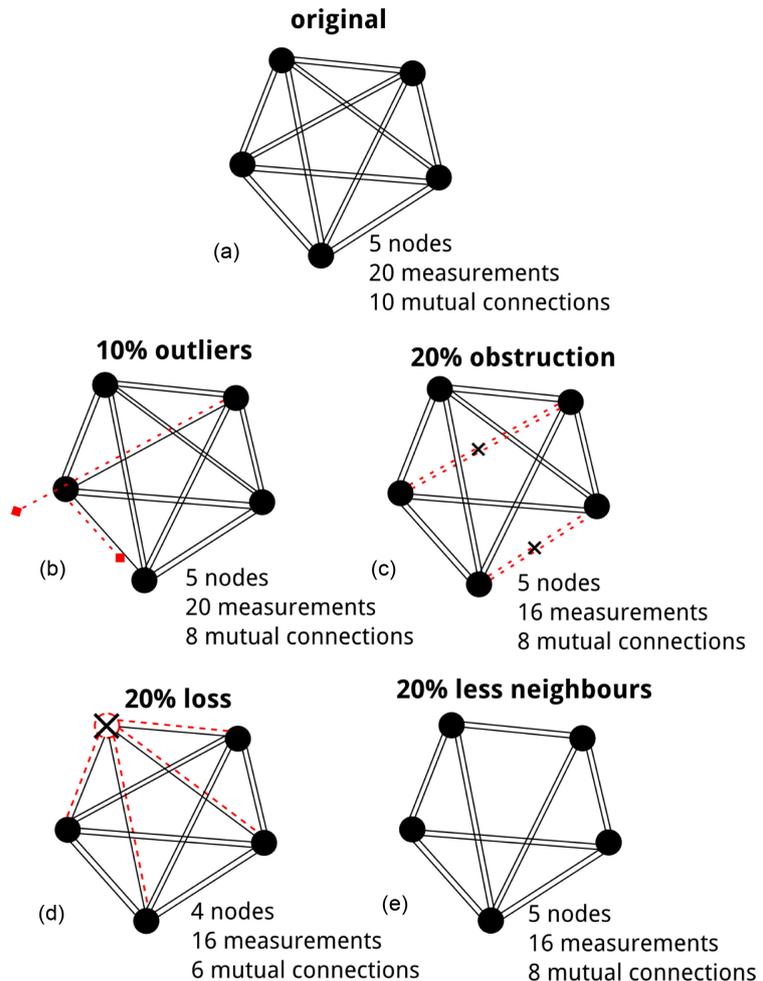
### 3.4. Mutual Connections

When the nodes are retrieved after the experiment and their data read out, offline analysis can be performed. Normally, with unique identification, mutual connections can be established by associating measurements  $d_{i,j}$  and  $d_{j,i}$  with each other (**Figure 3(a)**). These mutual connections can be used as consistency check of the individual measurements and quantify the uncertainty of the measured distance. When the two individual measurements differ less than a specific threshold, the measurements can be considered *inliers* and a mutual agreed-upon distance can be established, otherwise they are considered *outliers* and cannot be used to establish a mutual connection.

However, in the case of non-unique identification, measurement  $d_{i,\text{CID}(j)}$  (**Figure 3(b)**) can only be used correctly when the measured non-unique identifier  $\text{CID}(j)$  is associated with node  $j$ . Obtaining the mutual connection where  $d_{i,\text{CID}(j)}$  and  $d_{j,\text{CID}(i)}$  are associated with the correct nodes  $j$  and  $i$ , respectively (similar to **Figure 3(a)**), is not trivial as the mapping from CID to UID is ambiguous.

Mutual connections, where  $d_{i,CID(j)}$  is associated with  $d_{j,CID(i)}$ , are obtained by applying a consistency check on each of the possible node pairs that have cids consistent with the measurements. A mutual connection is hypothesised when two distance measurements agree on two grounds: 1) the measured cids agree with the CID of the other, and; 2) the corresponding measured distance is approximately similar to the other (*i.e.* inlier measurement). The exact implementation is described in Section 4.1.

This consistency check is affected by the different types of measurement noise in the individual measurements, specifically the outlier-type of noise that prevent finding the correct mutual connections. **Figure 4** illustrates how the



**Figure 4.** An example of (a) fully connected graph with five nodes; (b) outlier noise for two measurements; (c) obstruction between two nodes; (d) loss of one nodes; and (e) reduction of communication radius. Identification noise can be seen as outlier noise but on identification instead of distance. Explanation: Nodes are indicated as dark circles; a line indicates a distance measurement between two nodes. A too long/short line indicates a measurement with outlier noise. A dashed line indicates the distance measurement has never existed (obstruction) or this data is not retrieved (loss).

different types of outlier-type noise affect finding correct mutual connections. **Figure 4(a)** shows a fully connected reference graph in which all mutual connections are present and correct. Outlier noise in the distance measurements, as shown in **Figure 4(b)**, leads to a set of  $d_{i,CID(j)}$  and  $d_{j,CID(i)}$  measurements that cannot be associated with each other as the difference in measured distance is larger than the inlier threshold. Identification noise can be seen in a similar way with the difference being that instead of erroneously measured distances the identifier CID is decoded erroneously. Obstruction of signal paths (**Figure 4(c)**) and the inability to retrieve nodes (**Figure 4(d)**) also prevents establishing mutual connections. From a reconstruction perspective, not being able to establish mutual connections due to the presence of noise can to some extent be compared to reducing the number of neighbouring nodes (reducing the communication radius), which is visualized in **Figure 4(e)**.

## 4. Swarm Reconstruction Algorithm

In this section we will describe the novel reconstruction algorithm. It reconstructs node positions based on inter-node distances with significant measurement imperfections and the ambiguities introduced by using non-unique identifiers.

The reconstruction algorithm can be summarized as follows:

- 1) Hypothesise mutual connections based on the consistency (distance and identity) between all distance measurements, obtained from all retrieved nodes (Section 4.1);
- 2) Find initial four nodes to fix coordinate reference system (Section 4.2);
- 3) Robustly add additional nodes to the graph using a guided-RANSAC algorithm and general lateration, exploiting the geometric consistency between true neighbouring nodes (Section 4.3);
- 4) Perform non-linear optimization to reduce build-up of errors from previous step; go back to step 3 (Section 4.4);
- 5) Loop closing, if needed (Section 4.5).

An overview of the algorithmic steps of the reconstruction method is provided in **Appendix A**.

### 4.1. Identity and Consistency Check

Before the reconstruction process, mutual connections should be established between nodes. The decoded non-unique communication identifier, *cid*, from every received ranging message should be associated with a unique (hardware) identification number, UID. Initially, this can be approached using a combinatoric method in which all possible nodes are considered that use the respective *cid*. The range of options can be narrowed down, as the distance measurement  $d_{i,CID(j)}$  of node *i* to node *j*, when performed correctly and without noise, should be similar to the reverse distance measurement  $d_{j,CID(i)}$  of node *j* to node *i*. This means that searching the measured data for the consistency where  $d_{i,CID(j)}$  is similar to  $d_{j,CID(i)}$  should give most of the times an unambiguous mutual

correct connection, since it is unlikely that another set of nodes with similar CID pair have the same matching distance.

However, the distance measurements are imperfect and noisy. Consequently, the distance measurements and the “reverse” distance measurement will not be equal. Therefore, potential mutual connections are hypothesised when the measured distances lie within a specific threshold value  $\epsilon_r$  of each other and the measured identifiers are consistent with each others cid. This condition, denoted with  $d_{a,b} \approx d_{c,d}$ , then becomes:

$$d_{a,b} \approx d_{c,d} \quad \text{when} \quad \begin{cases} |d_{a,b} - d_{c,d}| \leq \epsilon_r \\ \text{CID}(a) = \text{CID}(d) \\ \text{CID}(b) = \text{CID}(c) \end{cases} \quad (2)$$

For all pairs that are considered consistent with each other, *i.e.*  $d_{a,b} \approx d_{c,d}$ , a mutual connection is hypothesised and the average distance is taken as mutually agreed upon distance between them. All other pairs are not considered a mutual connection. The final consistency function  $\mathcal{C}$  is expressed as:

$$\mathcal{C}(d_{a,b}, d_{c,d}) = \begin{cases} \frac{d_{a,b} + d_{c,d}}{2} & \text{when } d_{a,b} \approx d_{c,d} \\ \emptyset & \text{otherwise} \end{cases} \quad (3)$$

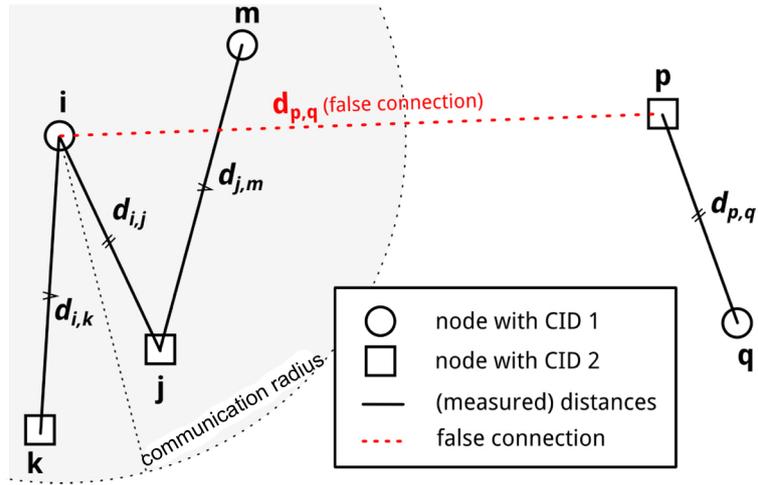
where  $\emptyset$  indicates that no mutual connection is considered, *i.e.* an empty entry.

Threshold value  $\epsilon_r$  is the maximum allowed difference in distance measurement for which two measurements are considered inliers. This threshold should be related to the noise. It can e.g. be obtained using trial and error if the error model is unknown. Note that due to the measurement noise, the threshold, and the non-unique identification not all pairs that are considered consistent are correct, they are only hypothesised as being correct connections.

The obtained mutual connections between node  $\bar{i}$  and node  $\bar{j}$  and their hypothesised distance is not unambiguous. As illustrated in **Figure 5**, the hypothesised mutual connections  $h_{i,j}$  for measurement  $d_{i,\text{CID}(j)}$  can consist of three types of contributions:

$$h_{i,j} = \begin{cases} \mathcal{C}(d_{\bar{i},\bar{j}}, d_{\bar{j},\bar{i}}) & \text{(a)} \\ \mathcal{C}(d_{\bar{i},j}, d_{p,q}) & \text{(b)} \\ \mathcal{C}(d_{\bar{i},k}, d_{j,m}) & \text{(c)} \end{cases} \quad (4)$$

- 1) the measurement between the real nodes  $\bar{i}$  and  $\bar{j}$ ;
- 2) *identity ambiguity*: a mutual connection to a node  $p$  which is believed to be at similar distance as node  $\bar{j}$  but not necessarily within communication radius of  $\bar{i}$  (see **Figure 5**);
- 3) *distance ambiguity*: a mutual connection with the correct node  $\bar{j}$  but with a distance belonging to the measured distance to another node  $k$  within its communication radius (see **Figure 5**).



**Figure 5.** Nodes with  $UID = \{i, j, k, m, p, q\}$  using only two distinct cids (indicated by  $\circ$  and  $\square$ ) cause ambiguities in the consistency check. Identity ambiguity arises due to *similar* distance between pairs with similar CID in the swarm (Equation (2)). Distance ambiguities arise due to a plurality of nodes with similar cids within communication radius.

The identity ambiguities are based on the statistical likelihood that somewhere in the swarm, a node pair with similar cids and distance is present such that the measurement pair obeys Equation (2). Assuming a uniform spatial distribution of the nodes, the average number of identity ambiguities in  $h_{i,j}$  scales with  $\mathbb{A}_I \propto \epsilon_r n_{\text{comm}} N / n_{\text{CID}}^2$ . Equally, the distance ambiguities are based on the statistical likelihood that neighbouring nodes give rise to confusion. The average number of distance ambiguities per  $d_{r, \text{CID}(j)}$  then scales with  $\mathbb{A}_D \propto \epsilon_r n_{\text{comm}}^2 N / n_{\text{CID}}^2$ .

Both type of ambiguities,  $\mathbb{A}_I$  and  $\mathbb{A}_D$ , increase with an increasing number of neighbouring nodes  $n_{\text{comm}}$  and a lower  $n_{\text{CID}}$ , leading to a more challenging task deciding which are correct mutual connections and which are false connections. These ambiguities are resolved in our RANSAC graph-growing algorithm as detailed in Section 4.3.

### 4.2. Initial Seed Selection

In this work we solve the reconstruction problem by incrementally growing the graph using a robust RANSAC-based method based on general lateration principles. Given an initial graph  $\mathcal{G}$ , new nodes can be added as illustrated in **Figure 2** and detailed in [6]. In an ideal case (without noise), every candidate node,  $c$ , of which the distances to four non-coplanar nodes with known positions  $s_1, s_2, s_3, s_4$  are known, can be added to the graph with as position the intersection of the spheres with radii  $\bar{d}_{1,c}, \bar{d}_{2,c}, \bar{d}_{3,c}, \bar{d}_{4,c}$  and centers at  $s_1, s_2, s_3, s_4$ . Here  $\bar{d}_{n,c}$  denotes the true distance from node  $n = \{1, 2, 3, 4\}$  to node  $c$ .

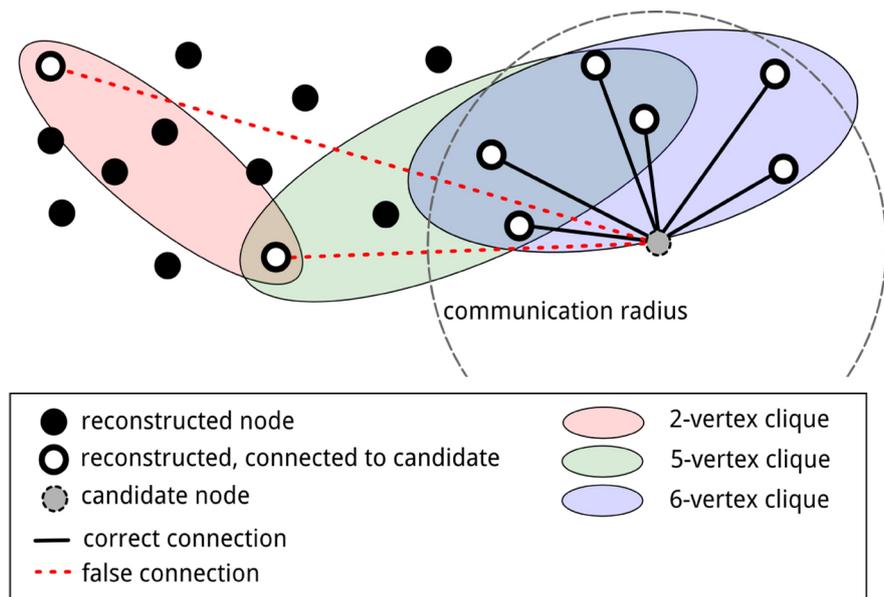
Since the graph growing is performed in  $\mathbb{R}^3$  and the distance measurements are performed in  $\mathbb{R}$  and without external beacons acting as fixed reference

points in space, first a coordinate reference system has to be defined. The initial four node positions are chosen such that they define the coordinate system, therewith resolving the general reflection and rotation ambiguity. Their positions,  $s_1, s_2, s_3, s_4$ , define the coordinate system as follows, and also illustrated in **Figure 2**:  $s_1 \in \{0, 0, 0\}$ ,  $s_2 \in \{\mathbb{R}^+, 0, 0\}$ ,  $s_3 \in \{\mathbb{R}, \mathbb{R}^+, 0\}$  and  $s_4 \in \{\mathbb{R}, \mathbb{R}, \mathbb{R}^+\}$ . These nodes are selected based on their connectivity and the stability of their geometric configuration which is obtained by general lateration techniques.

Since  $h_{i,j}$  contains ambiguities, the initial seed selection is conditional until the graph growing has successfully added several nodes to the graph  $\mathcal{G}$ . If this is not possible, a new initial seed is selected and the process is repeated.

### 4.3. RANSAC Graph Growing

After the initial seed is chosen, additional nodes can be added to the graph when the nodes have at least four connections to already reconstructed nodes. The set of candidate nodes that have at least four connections to already reconstructed nodes is denoted as  $\mathcal{C}$ . Ideally,  $\mathcal{C}$  only consists of nodes that are true neighbouring nodes, but due to the ambiguities in the hypothesised connections  $h_{i,j}$  this is not the case. **Figure 6** illustrates a simplified 2-D situation in which a candidate node  $c \in \mathcal{C}$  has connections to its true neighbouring nodes in  $\mathcal{V}$ , but also false connections to nodes somewhere else in the graph due to identity ambiguities. Whether the connections are true or false is not known at this point.



**Figure 6.** RANSAC graph growing algorithm uses an inlier-outlier voting system to filter out outlier distance measurements and proposes a position for candidate nodes to add to the graph. Using cliques of nodes in the graph that are within twice the communication radius helps in reducing the probability that nodes are positioned based on false connections.

A Random Sampling Consensus (RANSAC) method is used to attempt to correctly position the candidate nodes. RANSAC is a general robust estimation technique, for more details we refer to [27]. RANSAC attempts to fit a model on data points that contain both inliers (points that satisfy the model) and outliers (points that do not satisfy the model). It does this by randomly selecting a small set of data points, estimating a model based on this small set, and counting the number of other data points that agree with this model. This process is repeated until a model is found that has maximum or sufficient support in the whole set of data points. We developed a specific RANSAC algorithm to solve the task of graph-growing under severe outlier noise and identification ambiguity. An overview of the RANSAC algorithm is provided in **Appendix B**.

When a candidate node  $c$  is considered, all nodes that are already reconstructed in the graph and that are connected with this  $c$  are selected. This set of nodes is denoted with  $A = \{a : a \in \mathcal{V}, h_{a,c} \neq \emptyset\}$ . A subset of three of these nodes is selected,  $A_p = \{a_p \in A\}$ , to *propose*—using general lateration—a position for  $c$  up to a reflection ambiguity (three nodes are always coplanar). However, no guarantees can be given whether the entries in  $h_{a_p,c}$  are correct connections and the proposed positions do not need to be viable or close to the true position. Therefore, each of the other connected nodes

$A_v = \{a_v : (A_v \cup A_p) = A, (a_v \cap a_p) = \emptyset\}$  are used to *vote* for the proposed position. Consensus is reached when the majority of the voting nodes, including the three proposing nodes, have at least a 50% majority. The reflection ambiguity is resolved by choosing which of the two positions received more supporting votes. When consensus is reached, the number of supporting votes is called the *RANSAC-score* for the specific proposal.

This step in the RANSAC algorithm is repeated with each time a different set of three proposers  $a_p$ , until all possibilities are exhausted or until a proposal received a specific threshold in RANSAC-score. In the latter case it is then considered sufficiently supported and is added to the graph. When no “sufficiently supported” condition is reached, the RANSAC procedure is repeated for a next candidate node until all candidates are considered or a sufficiently supported condition is found for a specific candidate. The candidate with the highest RANSAC-score is added to the graph in  $\mathcal{V}$  with edges ( $\mathcal{E}$ ) only to the supporting nodes and corresponding  $h_{a,c}$  entry.

The “sufficiently supported” condition for the RANSAC can be based on the expected percentage of outlier measurements or can be chosen heuristically by trial-and-error, as is done in this work.

### Guiding RANSAC

The number of false connections from the candidate nodes to the already reconstructed nodes, grows on average linearly with the number of nodes already in the graph, due to  $\mathbb{A}_l$ , as explained in Section 4.1. Consequently, when a candidate node is considered that does not have enough true neighbouring nodes in the graph yet but does have a large number of false

connections to nodes already in the graph, this candidate node might receive a large RANSAC-score but based on only false connections. The true neighbouring nodes have in that case not enough voting power over the number of false neighbouring nodes.

To reduce the probability of this to happen, the sampling procedure of the candidate nodes is guided using a sorting order of candidate nodes before the RANSAC algorithm is performed on them. The sorting is based on the likelihood that candidate nodes have enough true neighbouring nodes already in the graph. As such, candidate nodes that are more prone to be reconstructed based on false connections, are considered later. This gives the probability that more of its true neighbouring nodes will be added to the graph first. The algorithm is provided in **Appendix C** and described next.

For each candidate node  $\{b: b \in \mathcal{C}\}$ , a list is made of connected nodes that have already been reconstructed in the graph  $\{a: a \in \mathcal{V}, h_{a,b} \neq \emptyset\}$ . A new graph  $\mathcal{G} = (\mathcal{V}(a), \mathcal{E})$  is made with nodes  $a$  as vertices. These vertices are connected with edges  $\mathcal{E}'$  when their positions are less than twice the communication radius,  $2r_{\text{comm}}$ , apart. In this new graph, maximal cliques are listed as  $\mathcal{Q}(b)$ , and indicate the groups of nodes that are potentially within the communication radius of a proposed candidate node position. This means that on geometrical reasons, all of these nodes can potentially agree on candidate node position. In other words, nodes from outside this clique could never agree with all nodes within the clique on a candidate node position. The example in **Figure 6** shows these cliques in coloured oval areas. The number of nodes in these cliques are registered and are used as sorting order in which RANSAC is performed. Nodes with larger sized cliques in  $\mathcal{Q}(b)$  are more likely to have more connections with true neighbouring nodes in the graph.

This sorting has two effects on the reconstruction, it guides the RANSAC procedure and has as a result that: 1) it increases the probability that a “sufficiently supported” candidate is found quickly, and 2) it increases the probability that from all “sufficiently supported” candidates, the best one is chosen for addition to the graph.

Guiding the RANSAC procedure with this sorting requires additional computational power (calculating the cliques), relative to unguided RANSAC. This increase in required computational power is compensated for by the quicker finding of a “sufficiently supported” candidate. The reconstruction can roughly take between 0.1 - 10 seconds per node, depending mainly on the connectivity and ambiguities.

#### 4.4. Robust Non-Linear Refinement

The stepwise addition of new nodes to the graph introduces build-up of errors. These errors in positions can prevent other nodes from being added. In order to reduce this error build-up, a global non-linear optimizer algorithm is executed as described in [6] to minimize the cost function Equation (1).

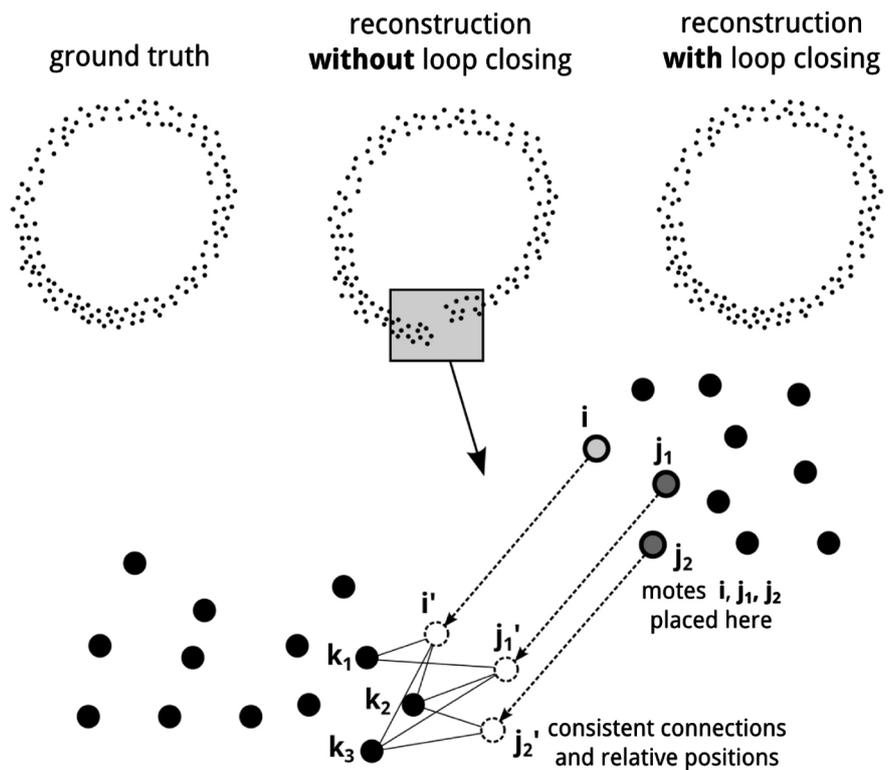
For this we use the efficient general graph optimization software package  $g^2o$  [28]. It uses sparse methods to solve the normal equations at the core of the non-linear optimization techniques like Levenberg-Marquardt or a Gauss-Newton. It can easily solve graph problems consisting of thousands of nodes and edges.

In our algorithm, after every  $m$  newly added nodes, or failure to add a new node this non-linear optimization is performed. There is a clear trade-off between  $m$  and the processing time required for the reconstruction algorithm. In our work,  $m$  is chosen arbitrarily to 10. In future work, this can e.g. be adjusted dynamically based on uncertainty of previously added nodes.

### 4.5. Loop Closing

The algorithm described above only considers edges between nodes in  $\mathcal{G}$  when the specific connections supported the winning proposal in the RANSAC voting. As can later be seen in the results section, Section 5, this is effective against any type of outlier noise. But when considering loops in the environment, these loops might not be closed due to the previously mentioned error build-up, as schematically illustrated in **Figure 7**.

Nodes that are supposed to connect both ends of the loop, will only be placed at one of both ends, depending on which has the largest number of supporters.



**Figure 7.** A 2-D example of an environment with a loop (ground truth). Due to error build-up in the RANSAC graph growing, loops are not closed (reconstruction without loop closing). Loop closing candidates can be detected by comparing local geometric consistency of excluded connections with reconstructed positions. After detection, the loop can be closed (reconstruction with loop closing).

Placement on the other end of the loop could also have yielded successful reconstruction, however, the node will not automatically be added to both ends to close the loop. An additional step should be performed in order to detect such loops and optimize the whole graph such that the loop is closed. This section describes the method how these loops are detected and closed.

Conceptually, the detection of loops is performed by searching for a set of nodes that could have been reconstructed in two different, distant, positions in the graph. This starts by looking at all hypothesised connections between nodes that are not used in the reconstruction: *i.e.* all connections  $h_{i,j}$  that did not support a winning RANSAC proposal, we call these *excluded connections*. The connections that did get included are called *included connections*.

Let node  $i$  have excluded connections to nodes  $k_{\text{ex}}^i$  and included connections to reconstructed neighbouring nodes  $j_{\text{in}}^i$ , as illustrated in **Figure 7**. In order to successfully detect and close a loop, sufficient connectivity should exist between nodes  $j_{\text{in}}^i$  and  $k_{\text{ex}}^i$ .

After the reconstruction has stopped or halted, for every node  $i$  it is checked how many connections are excluded from the reconstruction, and how many of the neighbouring nodes  $j_{\text{in}}^i$  also have excluded connections to nodes  $k_{\text{ex}}^i$ . When several neighbouring nodes agree that nodes are excluded, the node is said to be *commonly excluded*. Nodes  $k_{\text{ex}}^i$  that are commonly excluded by nodes  $j_{\text{in}}^i$  are likely to be on an other end of a loop than nodes  $j_{\text{in}}^i$ .

Final determination of the loop closing can be performed by redoing the RANSAC graph-growing on this set of connections by forcing the nodes to be added to the graph on the other side of the loop. When this yields a successful reconstruction, and the local relative positions of the considered nodes is equivalent to the local relative positions on the original side of the loop; then the loop can be closed. The loop closing itself can be performed by including the specific loop-closing edges that were initially excluded to the original graph and running the non-linear graph optimizer on the newly obtained graph.

## 5. Numerical Simulations

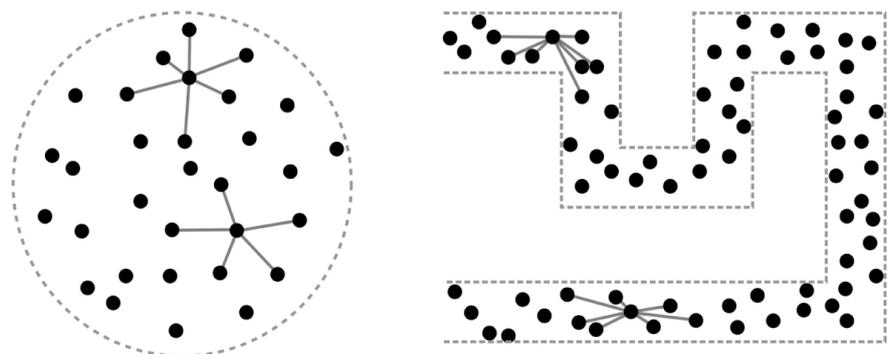
In this section we will use numerical simulations to study the performance of the reconstruction algorithm, as well as how the different soft constraints (connectivity and identification ambiguity) influence this performance. The simulations are performed using two vastly different environment geometries, nodes are given a position in a pipeline environment and in a spherical environment (Section 5.1). The distance measurements are generated based on the distances between these positions and a variety of noise types is added to account for different measurement imperfections (Section 5.2). These different noise types can influence the distance measurements such that the number of neighbouring nodes to which inlier measurements are available is reduced (Section 5.3). Assessing the performance of the reconstruction algorithm can be performed in different ways, depending on which qualities are favoured of the

reconstruction. Therefore, several performance metrics are introduced to address these different qualities (Section 5.4). All parameters related to the soft constraints are swept over a broad range until the reconstruction algorithm will fail to reconstruct the node positions (Section 5.5). Knowing the failure point of our reconstruction algorithm and understanding the trade-offs between the underlying constraints helps in guiding the hardware design for the development of future miniaturized nodes. The findings are summed up by plotting the performance against the calculated effective number of neighbouring nodes to see the most important trade-off (Section 5.6). In the last part, examples are given to show the loop closing detection and actual closing in environments that require this (Section 5.7).

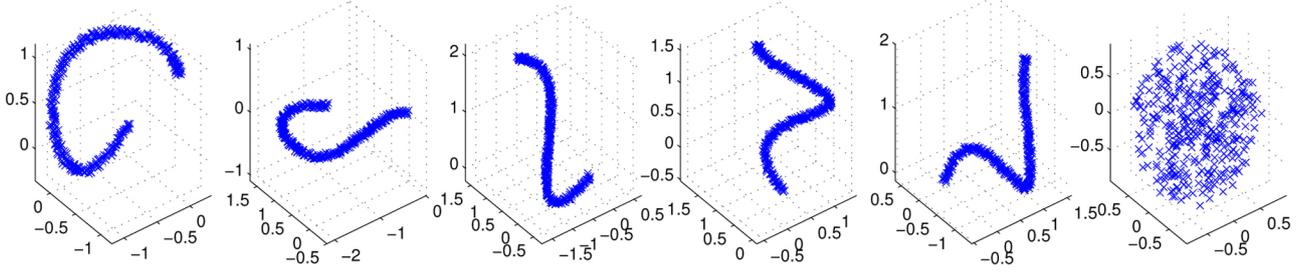
### 5.1. Environment Model

The envisioned node swarms can be deployed in a variety of environments. In this paper we will work with two types of environments as schematically illustrated in **Figure 8**. One resembles in abstract terms a mixing tank as the environmental dimensions are approximately the same in all directions. For convenience, the tank-like environment is chosen to be a bounded spherical environment. The second environment is a long pipeline, where the diameter of the pipeline is still large enough such that nodes can be positioned all around one another (as opposed to in one line along the pipe axis). The smooth pipeline has a fixed diameter of 8 cm and 400 nodes are placed in a section of 4 m pipe-length. The pipeline environment will be considered in both a loop and a loop-less fashion.

In the spherical environment, 400 nodes are given a random position within the boundary based on a uniform probability density function. The positions in the pipeline environment are chosen such that the nodes are spread out uniformly over the axis of the pipeline and the off-axis positions are chosen randomly based on a uniform probability density function over the cross-section. A total of 50 different spherical and 50 different pipeline environments are generated for the simulations. Examples can be seen in **Figure 9**.



**Figure 8.** A 2-D interpretation of the 3-D spherical environment and 3-D pipeline environment. The lines between the nodes indicate connectivity for two different nodes.



**Figure 9.** The ground truth positions of 400 nodes in several different experiments of randomly generated environments that are used for the simulations. Five pipeline environments, and one tank-like (spherical) environment.

In both environments, the communication radius  $r_{\text{comm}}$  of the nodes is chosen such that each node has on average  $n_{\text{comm}}$  neighbouring nodes. This parameter  $n_{\text{comm}}$  is one of the parameters that will be swept. The nodes communication radius  $r_{\text{comm}}$ , in which ranging measurements are possible, is for all nodes within the swarm the same and ranges throughout the experiments from  $10 \text{ cm} \leq r_{\text{comm}} \leq 15 \text{ cm}$  (corresponding to  $16 \leq n_{\text{comm}} \leq 28$ ).

## 5.2. Distance Measurement Generation

Distance measurements are generated based on the generated node positions. The distances between all nodes  $i$  and  $j$  as measured by nodes  $i$  are indicated by  $d_{\bar{i},j}$  and are modelled by the true distances between the nodes  $\bar{d}_{\bar{i},j}$  (the bar indicating true values).

Additive and multiplicative Gaussian are added to  $\bar{d}_{\bar{i},j}$  according to:

$$d_{\bar{i},j} = \bar{d}_{\bar{i},j} + \bar{d}_{\bar{i},j} \mathcal{N}_m + \mathcal{N}_a \quad (5)$$

where  $\mathcal{N}_m$  and  $\mathcal{N}_a$  being perturbations from the zero-mean Gaussian distributions  $\mathcal{N}(0, \sigma_m^2)$  and  $\mathcal{N}(0, \sigma_a^2)$  respectively.

Furthermore, a variety of operations are executed on the measurements to account for different outlier noise types, each with their own statistical likelihood,  $\nu$ , for this type of noise to happen to the measurement.

$$d_{\bar{i},j} = \mathcal{F}_{\text{identification}} \left( \mathcal{F}_{\text{outlier}} \left( \mathcal{F}_{\text{obstruction}} \left( \mathcal{F}_{\text{loss}} \left( d_{\bar{i},j} \right) \right) \right) \right) \quad (6)$$

with

$$\mathcal{F}_{\text{identification}}(d_{a,b}) = d_{a,b} \quad \text{when identification noise (probability } \nu_{\text{identification}} \text{)}$$

$$\mathcal{F}_{\text{outlier}}(d_{a,b}) = \mathcal{U} \quad \text{when outlier noise (probability } \nu_{\text{outlier}} \text{)}$$

$$\mathcal{F}_{\text{obstruction}}(d_{a,b}) = \emptyset \quad \text{when obstruction (probability } \nu_{\text{obstruction}} \text{)}$$

$$\mathcal{F}_{\text{loss}}(d_{a,b}) = \emptyset \quad \text{when node } a \text{ is lost (probability } \nu_{\text{loss}} \text{)}$$

and all operations return  $\mathcal{F}(d_{a,b}) = d_{a,b}$  when the specific types of noise do not occur. Identification and outlier noise occur randomly to each measurement, independently of each other. Obstruction and loss results in non-existing measurements, indicated with  $\emptyset$ . Obstruction happens to both  $d_{\bar{i},j}$  and  $d_{\bar{j},i}$

simultaneously; and loss happens to individual nodes, independently of each other.

Identification noise results in an erroneously decoded identifier, *i.e.* node  $j$  is measured as being node  $k$ . Outlier noise  $\mathcal{U}$  is chosen to be randomly drawn from a uniform distribution in the range  $(0, r_{\text{comm}}]$ .

Additionally, signal multipath can result in additional measurements of specific node pairs: besides the distance based on the direct signal path, also distances based on reflected, longer, signal paths. In order to account for this we add an additional distance measurement  $d'_{\bar{i},j}$  to the measurement dataset:

$$d'_{\bar{i},j} = \bar{d}_{\bar{i},j} + \mathcal{W} \quad (7)$$

with  $\mathcal{W}$  a perturbation drawn from a uniform distribution in the range  $(0, r_{\text{comm}})$ . The probability of this reflection happening is defined as  $\nu_{\text{reflection}}$ . For this reflection to be measured by node  $i$ , this distance should fall within the communication radius  $r_{\text{comm}}$ . A similar entry  $d'_{j,i}$  is added to account for the inverse measurement. Any reflection  $d'_{\bar{i},j}$  will also be subject to other discussed noise types.

This elaborate process generates realistic distance measurements and is significantly more extensive than that of previous research [6] [26] [29].

### 5.3. Effective Connectivity

The discussed noise not only affects the accuracy of the distance measurements but also determines whether correct mutual connections can be estimated (Section 3.4). Due to noise, the effective number of neighbouring nodes to which correct distance measurements are made is therefore lower than the number of neighbouring nodes  $n_{\text{comm}}$  that are within communication radius.

The *effective connectivity*, EC, between nodes is defined by the number of neighbouring nodes to which inlier measurements (and thus correct mutual connections) are available for the reconstruction algorithm. An estimation of the effective number of neighbouring nodes can be made based on the theoretical likelihood that inlier measurements can be established in the presence of the mentioned outlier noise types:

$$\text{EC} = n_{\text{comm}} (1 - \nu_{\text{outlier}})^2 (1 - \nu_{\text{id}})^2 (1 - \nu_{\text{loss}}) (1 - \nu_{\text{obstruction}}) \quad (8)$$

with  $\nu$  indicating the probability that the specific outlier-type of noise are present in the individual measurements as defined in Section 5.2. The terms for outlier and identification noise are squared as these are effects happening to  $d_{i,j}$  and  $d_{j,i}$  independently from each other but both influence the ability to establish a correct bidirectional measurement.

The reconstruction algorithm is developed to be able to robustly filter out different noise types by relying on the geometric consistency between true neighbouring nodes and their measured distances to each other. The more inlier measurements that are available, the more robust the algorithm is. The effective connectivity EC is a predictor of the number of inlier measurements and

therefore can be used as predictor of reconstruction performance. The usage of EC as such a predictor is verified in Section 5.6.

## 5.4. Reconstruction Performance Analysis

In our applications, or more in general for swarm operations like these, it is not established what is the best method to assess the performance of the reconstruction algorithm, *i.e.* to assess the quality of the reconstructed swarm. It is highly dependent on what information is favoured from such a swarm. Conventional metrics might not be suitable as the goal of the applications might be different. When exploring a yet unknown and difficult-to-access environment, initially one might want to know the overall structure of the environment: e.g. a rough estimation of the local geometry and the overall shape of the total environment. Later, e.g. when also adding additional sensor information, one might be more interested in fine-grained local geometry where the absolute error over the entire swarm is less relevant. We therefore present different performance metrics that serve different goals.

### 5.4.1. Absolute Error

The absolute error is the mean squared absolute error of all reconstructed node positions relative to their ground truth positions. It is calculated as  $\mathbb{E}_{\text{abs}} = \sum_i^N \|\hat{s}_i - \bar{s}_i\|^2 / N$ , the sum over all  $N$  nodes where  $\hat{s}$  is the reconstructed position of the node and  $\bar{s}$  its ground truth position. In order to compare the ground truth positions with the reconstructed positions, a linear fit between the 3-D positions of the initial four seed nodes and their ground truth positions is performed to resolve the general rotation and reflection ambiguity.

The mean squared absolute error is a commonly used performance metric in reconstruction studies. A disadvantage of this metric is that when a reconstructed swarm exhibits error build-up, the absolute error between one side of the swarm and the other can be large, while the local errors can actually be small. In swarm reconstruction studies like this, this metric should therefore not be used as the only metric to assess reconstruction performance.

### 5.4.2. Relative Error

The relative error is the mean squared relative error of the reconstructed distances between nodes, relative to the ground truth distances. It is calculated as  $\mathbb{E}_{\text{rel}} = \sum_{i,j}^M \|\hat{d}_{i,j} - \bar{d}_{i,j}\|^2 / M$ , the sum over all  $M$  reconstructed distances, where  $\hat{d}$  is the reconstructed distance and  $\bar{d}$  the ground truth distance.

Unlike  $\mathbb{E}_{\text{abs}}$ , the metric  $\mathbb{E}_{\text{rel}}$  is not affected by build-up of errors. It assesses the reconstruction performance only on a local and relative scale.

### 5.4.3. Global Error

The global error  $\mathbb{E}_{\text{glob}}$  is defined similar as  $\mathbb{E}_{\text{abs}}$ , but includes the performing of a rigid transform (transformation and rotation) of the entire reconstructed swarm such that  $\mathbb{E}_{\text{abs}}$  is minimized.

This metric illustrates the reconstruction performance of the swarm better

than  $\mathbb{E}_{\text{abs}}$  as it is not based on a specific chosen seed from which everything is built on. However, it is still sensitive to error build-up.

#### 5.4.4. Local Error

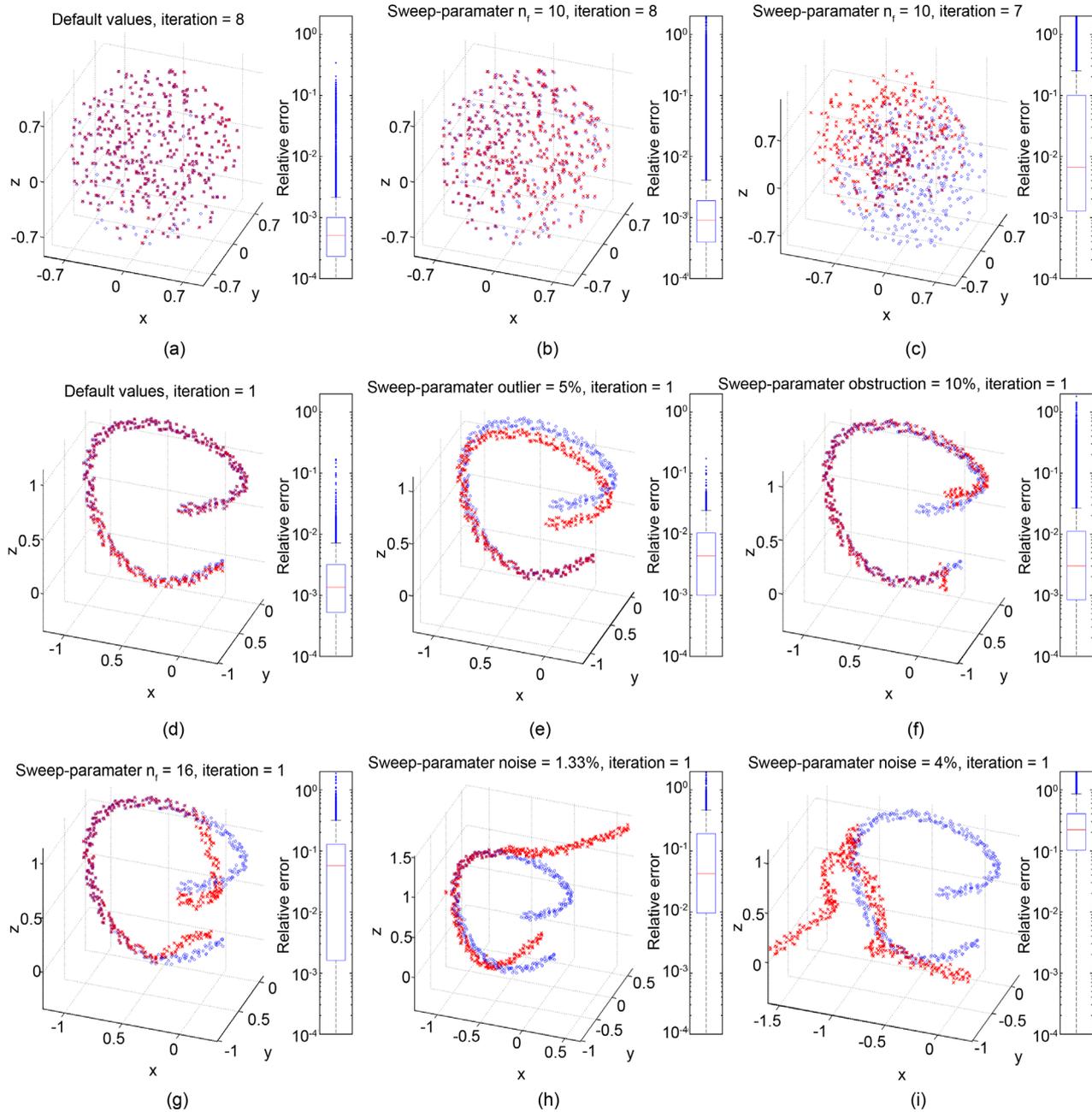
The local error  $\mathbb{E}_{\text{loc}}$  is defined as mean squared absolute error, but only after performing a rigid transform of all subsections of 20 connected nodes with their ground truth positions.

This heuristically chosen performance metric considers only the local errors, but other than  $\mathbb{E}_{\text{rel}}$ , it does focus on the (absolute) reconstructed positions of the nodes rather than the (relative) distances between them.

#### 5.4.5. Recall

The *recall*,  $\mathbb{X}$ , is the percentage of nodes reconstructed by the reconstruction algorithm. As this does not take into account whether or not these nodes are reconstructed correctly, we introduce the adjusted recall  $\mathbb{Y}$  and  $\mathbb{Z}$ . These are the percentages of nodes that are reconstructed within a specified error condition that is heuristically chosen. For  $\mathbb{Y}$ , this condition is when at least 80% of the node's reconstructed distances (or edges in  $\mathcal{G}$ ) have a relative error  $<10\%$ ; or when more than 50% of the nodes reconstructed distances have a relative error of  $<1\%$ . The adjusted recall  $\mathbb{Z}$  is similar to  $\mathbb{Y}$ , but includes the condition that the neighbouring nodes to which the relative error suffices this condition, should also fall in the category of  $\mathbb{Y}$ . Note that this categorization of recall into  $\mathbb{X}$ ,  $\mathbb{Y}$  and  $\mathbb{Z}$  is based on a heuristic, subjective interpretation of the reconstruction result.

Examples of reconstructions are shown in **Figure 10** and the corresponding performance metrics are listed in **Table 2**. The distribution of the relative errors of the reconstructed distances is indicated with a boxplot on the right of each reconstruction. Only **Figure 10(d)** and **Figure 10(a)** show reconstructions with a very low  $\mathbb{E}_{\text{abs}} \leq 1.4e-4 \text{ m}^2$  and visually seem like a perfect reconstruction. All others seem to exhibit their own erroneous characteristics, but non of them are completely wrong. **Figure 10(e)** has a fairly high  $\mathbb{E}_{\text{abs}}$  but  $\mathbb{E}_{\text{glob}} = 4.7e-4 \text{ m}^2$  with the same order of magnitude as  $\mathbb{E}_{\text{abs}}$  in **Figure 10(d)**. The reconstruction only seems to suffer slight error build-up that leads to large absolute errors at the pipe end but for the rest has an accurate reconstruction. For the majority of the swarm in **Figure 10(f)**, the positions are accurately reconstructed. Only at the ends of the pipe the reconstruction algorithm seems to have created large errors. In **Figure 10(h)** and **Figure 10(g)** the reconstructions also seems correct on the local scale (e.g. seen in the relative low  $\mathbb{E}_{\text{loc}}$ ) but exhibits some local errors that cause the reconstructed graph to deviate from the original axis of the pipeline. These differences illustrate the importance of having different error metrics as the assessment should happen based on which information is favoured from the swarm. As an extreme, even in **Figure 10(i)** the diameter of the pipe can still be estimated from the reconstruction while the shape of the swarm is very inaccurate.



**Figure 10.** Reconstructions of node swarms in a spherical environment (a)-(c) and in a pipeline environment (d)-(i) with different parameter sets. Blue circles are the ground truth positions of the nodes and red crosses the reconstructed positions. The recall metrics  $\mathbb{X}, \mathbb{Y}, \mathbb{Z}$  and the error metrics  $\mathbb{E}_{\text{abs}}, \mathbb{E}_{\text{rel}}, \mathbb{E}_{\text{glob}}, \mathbb{E}_{\text{loc}}$  are listed in **Table 2**.

Deciding whether a reconstruction gives satisfactory results or not is therefore not trivial and depends on the application goals. The adjusted recall parameters are an attempt to quantify these goals. In this paper, on the basis of subjectively interpreting how well the shape of the swarms in **Figure 10** are reconstructed, recall parameter  $\mathbb{Y}$  can be seen as a reasonable metric to assess the performance of the reconstructions. We can define a satisfactory reconstruction to be one with e.g.  $\mathbb{Y} > 80\%$ .

**Table 2.** Performance metrics of reconstructions shown in **Figure 10**.

	$\mathbb{X}(\%)$	$\mathbb{Y}(\%)$	$\mathbb{Z}(\%)$	$\mathbb{E}_{\text{abs}}(\text{m}^2)$	$\mathbb{E}_{\text{rel}}$	$\mathbb{E}_{\text{glob}}(\text{m}^2)$	$\mathbb{E}_{\text{loc}}(\text{m}^2)$
<b>Figure 10(a)</b>	97	97	97	$5 \times 10^{-5}$	$3 \times 10^{-5}$	$3 \times 10^{-5}$	$5 \times 10^{-5}$
<b>Figure 10(b)</b>	95	93	93	$1 \times 10^{-2}$	$2 \times 10^{-2}$	$1 \times 10^{-2}$	$2 \times 10^{-2}$
<b>Figure 10(c)</b>	88	64	0	$9 \times 10^{-1}$	$6 \times 10^{-2}$	$5 \times 10^{-1}$	$7 \times 10^{-2}$
<b>Figure 10(d)</b>	100	100	100	$1 \times 10^{-4}$	$2 \times 10^{-5}$	$5 \times 10^{-5}$	$4 \times 10^{-7}$
<b>Figure 10(e)</b>	99	99	99	$2 \times 10^{-2}$	$1 \times 10^{-4}$	$5 \times 10^{-4}$	$4 \times 10^{-7}$
<b>Figure 10(f)</b>	99	94	94	$6 \times 10^{-3}$	$5 \times 10^{-3}$	$4 \times 10^{-3}$	$2 \times 10^{-4}$
<b>Figure 10(g)</b>	99	59	0	$2 \times 10^{-2}$	$1 \times 10^{-2}$	$2 \times 10^{-2}$	$6 \times 10^{-4}$
<b>Figure 10(h)</b>	100	20	0	$3 \times 10^{-1}$	$5 \times 10^{-2}$	$2 \times 10^{-1}$	$1 \times 10^{-3}$
<b>Figure 10(i)</b>	99	0	0	$1 \times 10^0$	$3 \times 10^{-1}$	$2 \times 10^{-1}$	$5 \times 10^{-3}$

### 5.5. Parameter Sweeps and Breaking Points

In this section we study the performance of the reconstruction algorithm while sweeping the parameters related to the soft constraints. These parameters are the ones involved with measurement noise as described in Section 3.2, the number  $n_{\text{CID}}$  of non-unique identifiers cid, and the number of neighbouring nodes within communication radius  $n_{\text{comm}}$ . In order to change the number of neighbouring nodes we actually change the sensing radius  $r_{\text{comm}}$  such that on average, nodes have  $n_{\text{comm}}$  neighbouring nodes.

For each set of parameters, 50 different experiments are performed, each with randomly generated node positions (but following the environment geometry constraints as defined in Section 5.1). All experiments are performed with a total number of  $N = 400$  nodes. For similar work with a smaller and larger number of nodes, the reader is referred to our earlier work [6] and [7].

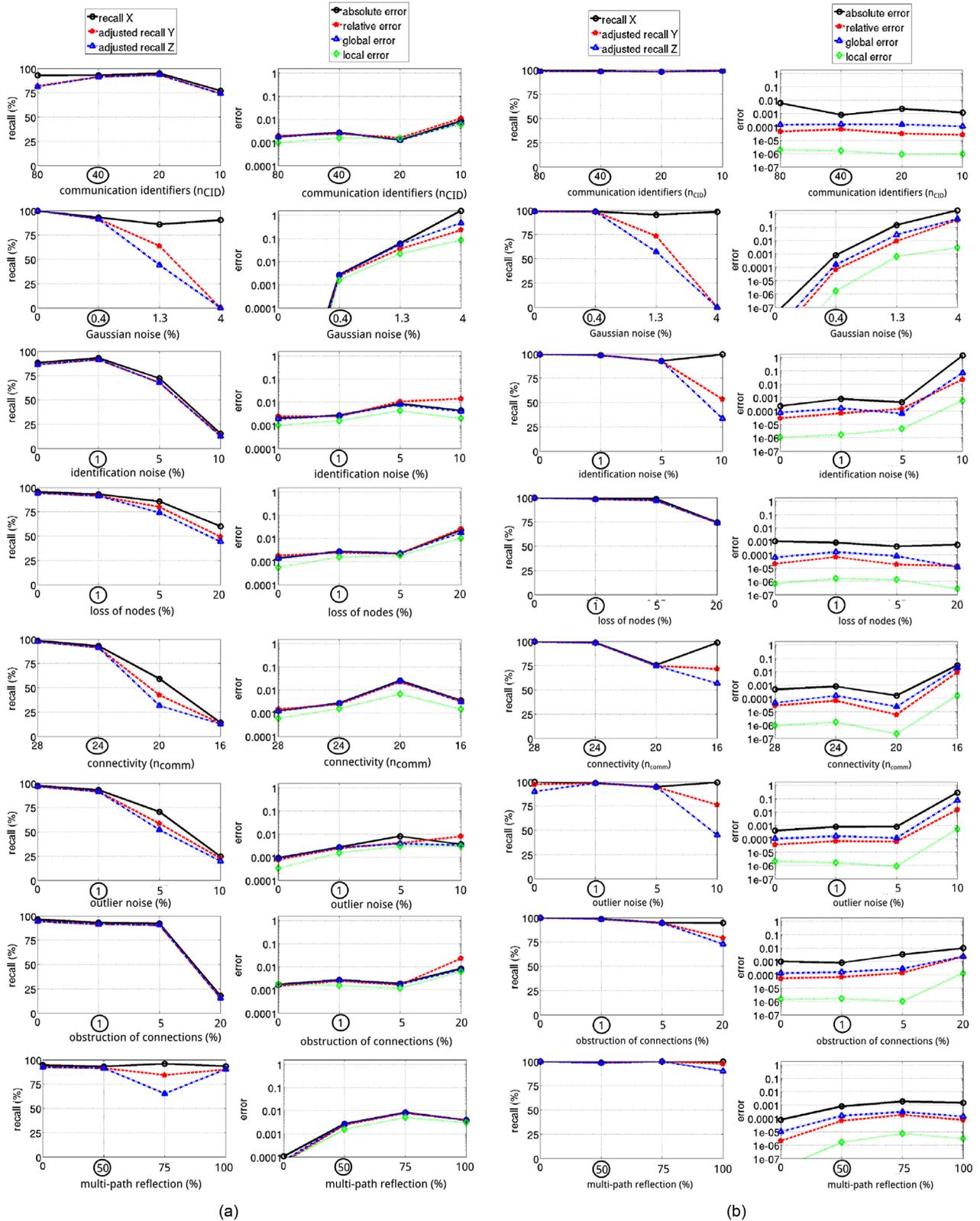
The Gaussian noise components in  $d_{\bar{i},\bar{j}}$  are together summarized as  $\nu_{\text{gaussian}}$ , as standard deviation of  $(d_{\bar{i},\bar{j}} - \bar{d}_{\bar{i},\bar{j}})$ . This value is expressed as percentage of a fixed communication radius<sup>2</sup>:

$$\nu_{\text{gaussian}} = \sqrt{\frac{\sum_{i,j} (d_{\bar{i},\bar{j}} - \bar{d}_{\bar{i},\bar{j}})^2}{M-1}} \frac{1}{r_{\text{comm}} [n_{\text{comm}} = 20]} \quad (9)$$

**Figure 11(a)** shows for a spherical environment the recall ( $\mathbb{X}, \mathbb{Y}, \mathbb{Z}$ ) and the different error metrics  $\mathbb{E}_{\text{abs}}, \mathbb{E}_{\text{rel}}, \mathbb{E}_{\text{glob}}, \mathbb{E}_{\text{loc}}$  while sweeping the input parameters. The median values of these metrics of all 50 experiments is visualized. It can clearly be seen that when increasing the measurement noise (Gaussian, outlier, identification, obstruction and loss), the recall drops and the error metrics increase.

**Figure 11(b)** shows the performance metrics using similar input parameters, but in a pipeline environment. It can be seen that the recall metrics in this case are much higher than in the spherical environment. This is due to a different

<sup>2</sup>The average communication radius when  $n_{\text{comm}} = 20$ .



**Figure 11.** Median of the recall result  $\mathbb{X}, \mathbb{Y}, \mathbb{Z}$  on the left and error metrics  $\mathbb{E}_{abs}, \mathbb{E}_{rel}, \mathbb{E}_{glob}, \mathbb{E}_{loc}$  on the right. Indicated parameters on the x-axis are swept while keeping other parameters on a fixed value (encircled). From top to bottom these are:  $\{n_{CID}, V_{gaussian}, V_{identification}, V_{loss}, n_{comm}, V_{outlier}, V_{obstruction}, V_{reflection}\}$ .

distribution of connectivity among the nodes, as explained next in Section 5.6.

Furthermore, the error metrics in the pipeline environment show significantly more differences among them than in the spherical case. The local error is orders of magnitude lower than for example the absolute and global error. The environment stretches across larger distances and has smaller local dimensions compared to the spherical case; errors build up easier and cause a larger absolute error, even in cases where the local geometry is reconstructed correctly. This can clearly be seen in e.g. **Figure 10(e)** and **Figure 10(h)**.

The observed dip in recall in the spherical environment in **Figure 11(a)** when reducing the number of identifiers to  $n_{\text{CID}} = 10$  is mainly due to a self-imposed time constraint in the reconstruction algorithm (when average time per reconstructed node exceeds 10 seconds). The ambiguities in this dataset are around  $\mathbb{A}_{\text{total}} = 200\%$ . Reducing the number of identifiers further can still yield successful reconstruction but drastically increases the number of ambiguities and therewith the time required to calculate cliques as described in Section 4.

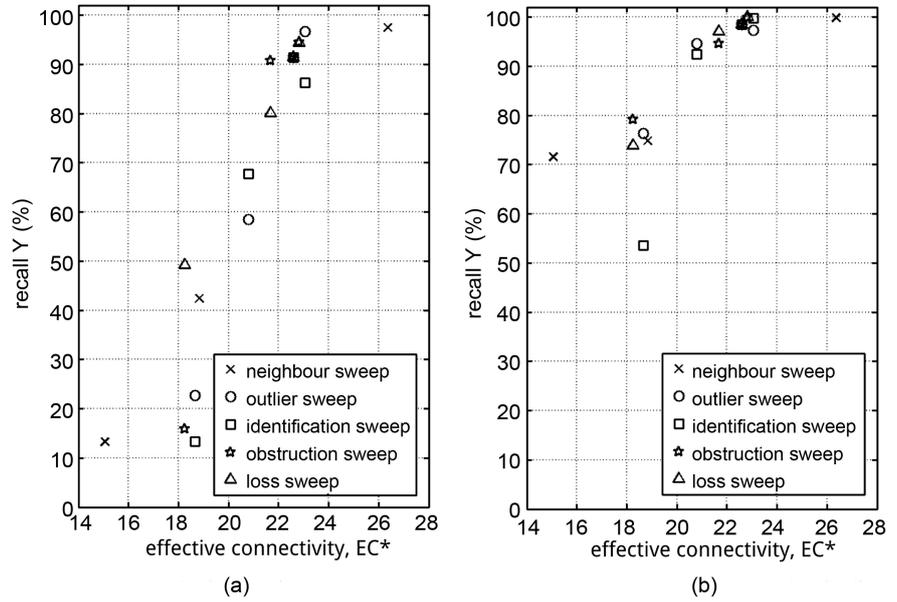
The lines in **Figure 11(a)** and **Figure 11(b)** only show the trend of the parameter sweep and due to the nature of the reconstruction process show statistical quirks. The standard deviation between all 50 experiments of the same input parameter sets are large and not shown to preserve readability of graphs. For example, in cases where the majority of experiments yield high recall (e.g. >90%), it is not uncommon that in one or more experiments of the same parameter set the recall does not surpass 10%.

The general trend of these lines clearly show there is a dependency of the input parameters on the reconstruction performance. The reconstruction exhibits graceful degradation when the soft constraints increase in severity up to the point where the recall drops and error increases.

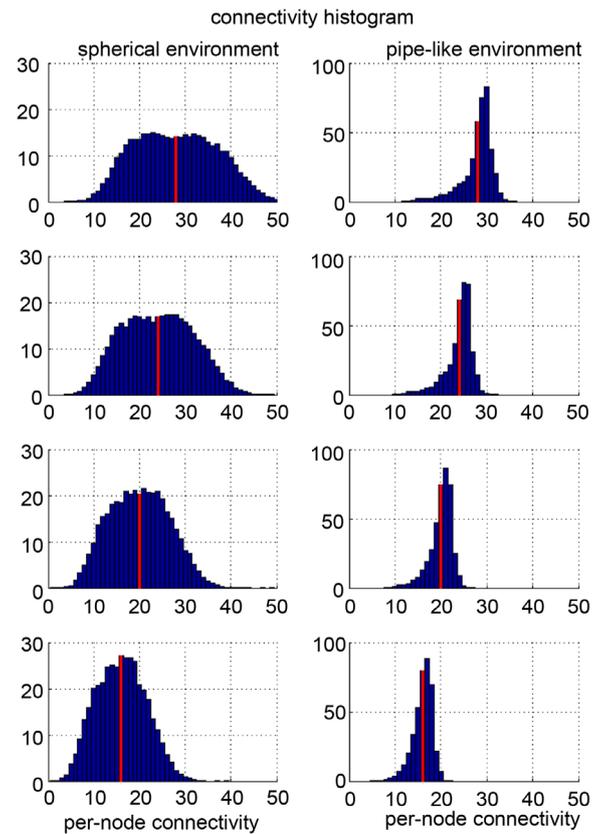
## 5.6. Effective Number of Neighbours

The recall result  $\mathbb{Y}$  of our experiments is plotted against the meta-parameter EC in **Figure 12**. For both environments, a clear trend is visible that the recall is dependent on the effective number of neighbouring nodes. A minimum number of inlier measurements is required that can be used for the RANSAC voting to guarantee satisfactory reconstruction of the nodes' positions. A minimum (effective) connectivity of around  $\text{EC} = 20 - 22$  is required in order to reconstruct the swarm with an adjusted recall  $\mathbb{Y} > 80\%$ .

Even though the average number of neighbouring nodes are chosen to be similar in both the spherical and the pipeline by setting  $r_{\text{conn}}$  for all nodes, the variability in the per-node connectivity in the spherical environment is much larger than in the pipeline environment. **Figure 13** shows the histogram of the per-node connectivity, averaged over all experiments. The chosen  $n_{\text{conn}}$  for each of the experiments is indicated using a red line, this is the average value of the histogram. The nodes with a lower connectivity are much less likely to be reconstructed (correctly) and will influence the recall result of the entire swarm.



**Figure 12.** Adjusted recall result  $\bar{Y}$  versus the meta-parameter  $EC$ , the effective number of neighbours. (a) Spherical environment; (b) pipeline environment.



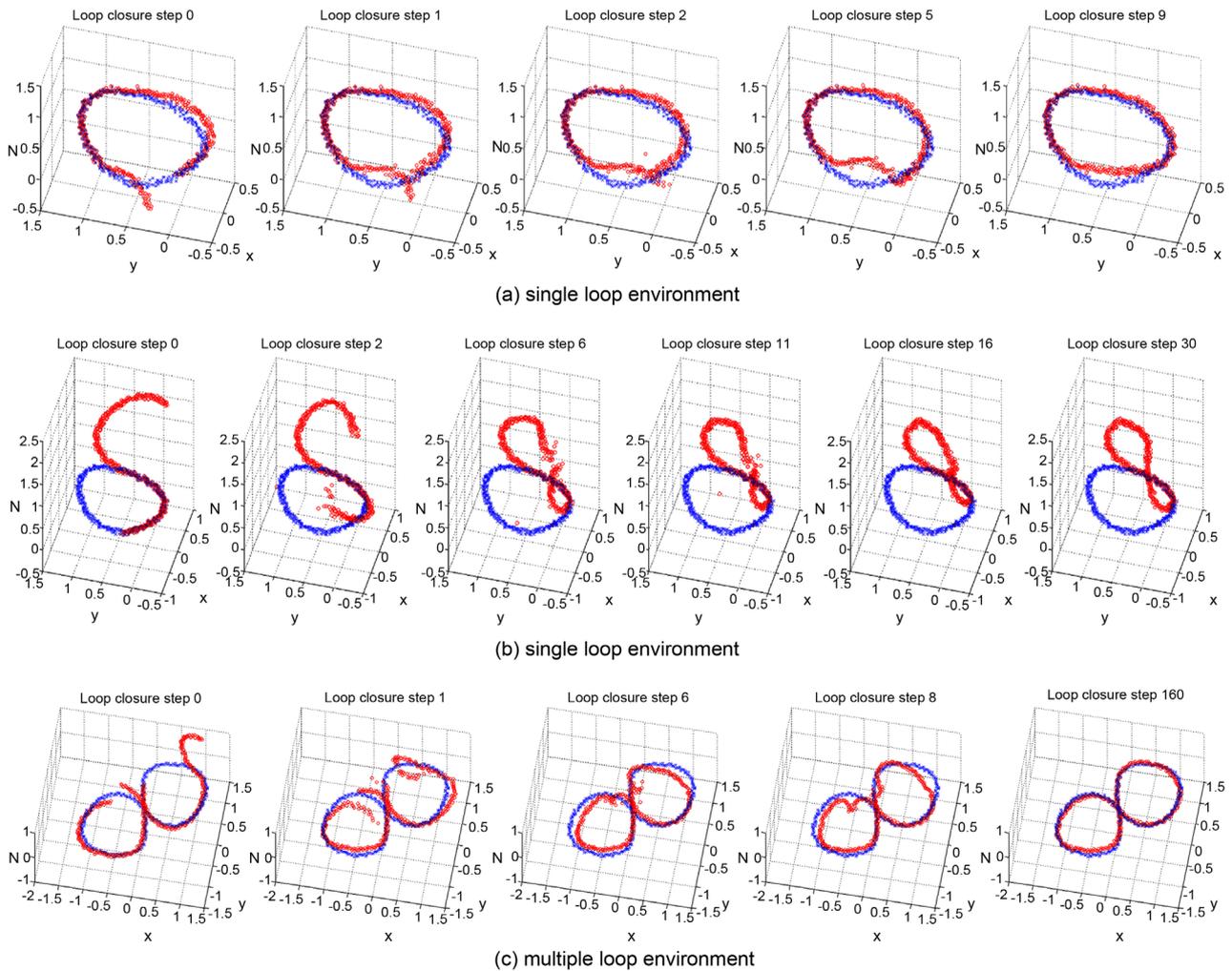
**Figure 13.** Histogram of real neighbouring nodes within communication radius. The average number of neighbours,  $n_{comm}$ , is indicated with a red line. From top to bottom  $n_{comm} = \{28, 24, 20, 16\}$ ; left the spherical environment right the pipeline environment.

It should be noted that the results in **Figure 11** and **Figure 12** show the median values over 50 experiments, each time with a different swarm topology and node positions. The findings presented here concern the reconstruction of the full swarm in a single reconstruction experiment. Smaller sub-swarms can be reconstructed using lower connectivity.

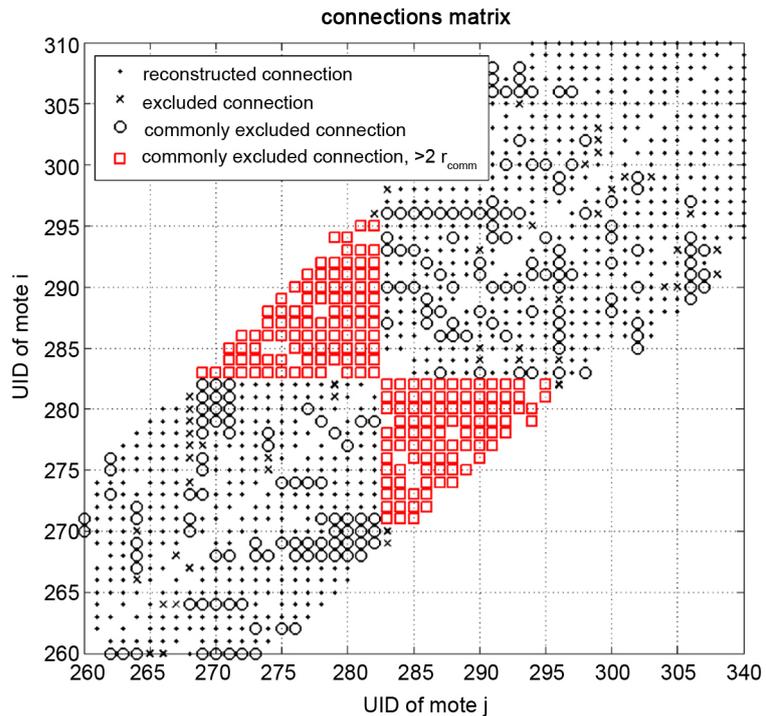
### 5.7. Loop Closing

Many environments will contain one or multiple loops; therefore it is important that the reconstruction algorithm can deal with closing these loops. Our loop closing method as described in Section 4.5 is evaluated in this section.

Examples of reconstructed swarms that require loop closing, and the iterative loop closing in the non-linear optimization process are shown in **Figure 14**. Build-up of error has caused the two ends of the loop(s) to not end up at the same position. **Figure 15** shows the result of the loop closing detection algorithm (Section 4.5) of the dataset shown in **Figure 14(b)**. The commonly



**Figure 14.** Several steps of the loop closing phase using non-linear optimization. Blue is ground truth positions and red is reconstructed positions.



**Figure 15.** This connection matrix shows reconstructed connections ( $\cdot$ ) and excluded connections ( $\times$ ). It indicates excluded connections that are excluded by a plurality of neighbouring nodes ( $\circ$ ) as well as those with a distance  $> 2r_{\text{comm}}$  ( $\square$ ). Loop closing candidates can easily be observed using this categorization. The linear ordering of the nodes' UID along the pipe axis is solely used for illustrative purposes. It is not used in the reconstruction as only cids are available from measurements.

excluded connections of which the reconstructed distance is larger than  $2r_{\text{comm}}$  are depicted with red squares and clearly indicate that the two ends of the reconstructed swarm are supposed to be connected. The nodes that are reconstructed at the loop-ends do not receive support from nodes on the other side of the loop in the RANSAC voting algorithm. To close the loop, these detected connections that are originally excluded from the graph are now included in the graph. Performing the non-linear optimization using these new connections, now closes the loop as seen in the consecutive steps in **Figure 14**.

As seen in **Figure 14(b)**, loop closing can distort the reconstructed swarm when the loop-ends are too far from each other. Closing the loop using this non-linear graph optimization then forces a different swarm geometry in such cases.

These experiments shows that our method can perform loop closing. Multiple loops can be dealt with simultaneously or individually, as seen in **Figure 14(c)**.

## 6. Discussion and Future Work

The results show that the reconstruction algorithm can robustly deal with a large set of different types of measurement noise. Each measurement imperfection has

a different influence on the established mutual connections and on the reconstruction. The inlier-type of measurement noise are present in all measurements and only affect the accuracy of the reconstructed positions. The outlier-type of noise prevents establishing correct mutual connections and on top of that also create false and ambiguous connections due to the non-unique identification. Using the geometric consistency of the positions of true neighbouring nodes, the reconstruction algorithm can robustly filter out the false connections and prevent them from negatively influencing the reconstruction. Reducing the quantity of available links significantly increases the ambiguities in establishing mutual connections. However, it is found that due to the robust nature of the reconstruction algorithm, these ambiguities could be resolved when enough inlier measurements are available.

The robustness of our novel algorithm allows us to see that the key parameter for successful reconstruction is the number of inlier node connections that are available for reconstruction. Initially, the total number of distance measurements is determined by the communication range and the density of the nodes in the swarm. It is due to the outlier-type of measurement noises that not all node connections will be inlier measurements and hence can not be used for reconstruction. The effective connectivity can be predicted by our Equation (8) and is an important predictor for the reconstruction performance. We can therefore conclude that in the sensor node hardware design for these applications, resources should be focussed on increasing the communication radius to increase connectivity and to prevent outlier-types of noise from reducing the effective connectivity. Achieving this can for example go at the cost of (unique) identification.

Depending on which information is favoured from the swarm, the error metrics can be used to study the performance of the reconstruction. There is no single error metric that can summarize the reconstruction performance for all application goals. Using different error metrics and the heuristically chosen adjusted recall metrics  $\Upsilon$  and  $\mathcal{Z}$  we showed that the subjective interpretation of the quality of the reconstruction can be partially quantified. When the sensor nodes need to be designed for specific applications, a performance condition can be set up, e.g.  $\Upsilon > 80\%$ , to assess under which input conditions the reconstruction is most likely to achieve the requested performance.

## 7. Conclusions

This work evaluated the feasibility of using sensor swarms, consisting of many highly miniaturized and severely resource limited sensor nodes, to reconstruct difficult-to-access environments. For this, a novel Guided Random Sample Consensus algorithm together with non-linear graph optimization is proposed. Its main contribution is that the algorithm can handle severe and different types of measurement errors as well as the use of ambiguous non-unique communication identifiers.

Extensive and realistic simulations show that sensor node connectivity is to be favoured over unique identification of nodes. When the number of used communication identifiers is only 2.5% of the total number of nodes ( $n_{\text{CID}} = 10$ ,  $N = 400$ ), the entire sensor swarm can still be reconstructed successfully. Thereby allowing less identification information to be used, *i.e.* shorter communication bursts, to trade-off for larger emission power per burst. This, in turn, allows for improved sensor node connectivity, as more nodes can be reached with a single burst. On the contrary, simulations show that when swarm connectivity drops below 20 inter-node connections per node, reliable sensor swarm reconstruction is hampered for this algorithm. Therefore, *favouring connectivity over identification*, is a pivotal finding of our work, as it can and will guide future developments of highly miniaturized sensor nodes.

## Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 665347.

## References

- [1] Talnishnikh, E., *et al.* (2015) Micro Motes: A Highly Penetrating Probe for Inaccessible Environments. In: Leung, H., Ed., *Intelligent Environmental Sensing*, Springer International Publishing, 33-49.  
[https://doi.org/10.1007/978-3-319-12892-4\\_2](https://doi.org/10.1007/978-3-319-12892-4_2)
- [2] European Union's Horizon 2020 FET-Open project: PHOENIX (No 665347).  
<https://www.phoenix-project.eu>
- [3] Patwari, N., *et al.* (2005) Cooperative Localization in Wireless Sensor Networks. *IEEE Signal Processing Magazine* (54), July.
- [4] Wymeersch, H., *et al.* (2009) Cooperative Localization in Wireless Networks. *P IEEE*, **97**, 427-450.
- [5] Duisterwinkel, H.A., *et al.* (2016) Asymmetric Multi-Way Ranging for Resource-Limited Nodes. *8th EAI International Conference on Ad Hoc Networks (AdHoc-Nets)*, Ottawa, 26-27 September 2016, 50-63.
- [6] Dubbelman, G., *et al.* (2014) Robust Sensor Cloud Localization from Range Measurements. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, 14-18 September 2014. <https://doi.org/10.1109/IROS.2014.6943099>
- [7] Duisterwinkel, H.A., *et al.* (2016) Mapping Swarms of Resource-Limited Sensor Motes: Solely Using Distance Measurements and Non-Unique Identifiers. *IEEE Symposium on Computational Intelligence for Engineering Solutions (SSCI 2016)*, Athens, 6-9 December 2016.
- [8] Jaffe, J.S., *et al.* (2017) A Swarm of Autonomous Miniature Underwater Robot Drifters for Exploring Submesoscale Ocean Dynamics. *Nature Communications*, **8**, Article Number: 14189 <https://doi.org/10.1038/ncomms14189>
- [9] Lu, J., *et al.* (2016) A Small Long-Life Acoustic Transmitter for Studying the Behavior of Aquatic Animals. *Review of Scientific Instruments*, **87**, 114902.  
<https://doi.org/10.1063/1.4967941>
- [10] Duisterwinkel, E.H.A., *et al.* (2017) Sensor Motes for the Exploration and Monitoring of Operational Pipelines. *IEEE Transaction on Instrumentation and Measure-*

- ments, **PP**, 1-12. <https://doi.org/10.1109/TIM.2017.2775404>
- [11] Wörtche, H.J. (2014) Advanced Analysis: Motes Dynamics in a Mixing Tank. Tech. Rep., INGU Solutions, Calgary, AB, Canada.
- [12] Durrant-Whyte, H. and Bailey, T. (2006) Simultaneous Localization and Mapping: Part I. *IEEE Robotics and Automation Magazine*, **13**, 99.
- [13] Djugash, J., *et al.* (2006) Range-Only SLAM for Robots Operating Cooperatively with Sensor Networks. *Proceedings 2006 IEEE International Conference on Robotics and Automation*, Orlando, 15-19 May 2006, 2078-2084. <https://doi.org/10.1109/ROBOT.2006.1642011>
- [14] Haehnel, D. (2004) Mapping and Localization with RFID Technology. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, 26 April-1 May 2004. <https://doi.org/10.1109/ROBOT.2004.1307283>
- [15] Neira, J., Davison, A. and Leonard, J. (2008) Guest Editorial, Special Issue in Visual Slam. *IEEE Transactions on Robotics*, **24**, 929-931.
- [16] Kirkham, R., *et al.* (2000) PIRAT—A System for Quantitative Sewer Pipe Assessment. *International Journal of Robotics Research*, **19**, 1033-1053. <https://doi.org/10.1177/02783640022067959>
- [17] Ribeiro, F., D., *et al.* (2011) Geometrically Constrained Room Modeling with Compact Microphone Arrays. *Audio, Speech, IEEE T AUDIO SPEECH*, **20**, 1449.
- [18] I. Dokmanić, *et al.* (2013) Acoustic Echoes Reveal Room Shape. *PNAS*, **110**, No. 30.
- [19] Menegatti, E., *et al.* (2009) Range-Only SLAM with a Mobile Robot and a Wireless Sensor Network. *IEEE International Conference on Robotics and Automation*, Kobe, 12-17 May 2009, 8-14.
- [20] Franchi, A., *et al.* (2013) Mutual Localization in Multi-Robot Systems Using Anonymous Relative Measurements. *The International Journal of Robotics Research*, May. <https://doi.org/10.1177/0278364913495425>
- [21] Akyildiz, I.F., Pompili, D. and Melodia, T. (2005) Underwater Acoustic Sensor Networks: Research Challenges. *Ad Hoc Networks*, **3**, 257-279. <https://doi.org/10.1016/j.adhoc.2005.01.004>
- [22] Duisterwinkel, E.H.A. (2018) Exploring Enclosed Environments with Floating Sensors: Mapping using Ultrasound. PhD Dissertation, Eindhoven University of Technology, Eindhoven, The Netherlands. (To be published)
- [23] Borg, I. and Groenen, P. (2005) Modern Multidimensional Scaling: Theory and Applications. 2nd Edition, Springer-Verlag, New York, 207-212.
- [24] Dagher, R., *et al.* (2014) Localization in Wireless Sensor Networks. In: *Wireless Sensor and Robot Networks from Topology Control to Communication Aspects*, Worldscientific, Location, 203-247.
- [25] Bachrach, J. and Taylor, C. (2005) Localization in Sensor Networks. In: *Handbook of Sensor Networks: Algorithms and Architectures*, John Wiley & Sons, Inc., Hoboken, 277-310.
- [26] Schlupkothen, S. and Ascheid, G. (2015) Localization of Wireless Sensor Networks With Concurrently Used Identification Sequences. In: *IEEE 14th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, Vilamoura, 17-18 June 2015. <https://doi.org/10.1109/MedHocNet.2015.7173166>
- [27] Raguram, R., Chum, O., Pollefeys, M., Matas, J. and Frahm, J.-M. (2012) USAC: A Universal Framework for Random Sampling Consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**, 2022-2038.

- [28] Kuemmerle, R., *et al.* (2011) g2o: A General Framework for Graph Optimization. *IEEE International Conference on Robotics and Automation*, Shanghai, 9-13 May 2011. <https://doi.org/10.1109/ICRA.2011.5979949>
- [29] Schlupkothen, S., *et al.* (2015) A Novel Low-Complexity Numerical Localization Method for Dynamic Wireless Sensor Networks. *IEEE Transactions on Signal Processing*, **63**, 4102-4114.

## Appendix A. Algorithm: Overview

**Data:**  $d_{\bar{i}, \text{CID}(j)}$ ,  $\text{CID}(\{1, \dots, N\})$   
**Result:** graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$   
 Consistency + hypothesise mutual connections  $h_{i,j}$  using  $\mathcal{C}$  (Sec. 4.1);  
 Create stable seed from 4 nodes (Sec. 4.2);  
**while** *unreconstructed nodes left* **do**  
   select all  $c \in \mathcal{C}$  that observe  $\geq 4$  nodes in  $\mathcal{V}$ ;  
   sort  $c$  according to most likelihood to succeed (Sec. 4.3.1);  
   perform RANSAC on  $c$  (Sec. 4.3 & Algo. 2);  
   **if** *nodes  $c$  with sufficient RANSAC-score* **then**  
     | add nodes with highest RANSAC-score to  $\mathcal{G}$ ;  
   **else if** *node added since last non-linear refinement* **then**  
     | run non-linear refinement (Sec. 4.4);  
   **else**  
     | abort algorithm;  
   **end**  
   **if**  *$m$  new nodes have been added to  $\mathcal{G}$*  **then**  
     | run non-linear refinement (Sec. 4.4);  
   **end**  
   Loop closing (Sec. 4.5)  
**end**

**Algorithm 1.** Reconstruction.

## Appendix B. Algorithm: RANSAC

**Data:** sorted  $c'$ ,  $h_{i,j}$ ,  $\mathcal{Q}$   
**Result:** RANSAC-score and proposed positions of  $c'$  in  $\mathcal{G}$   
**while** *no good enough RANSAC-score* **do**  
   select (next) largest clique in  $\mathcal{Q}$  with connected candidate  $b$ ;  
   select all possible sets of 3 nodes  $\{a : a \in \mathcal{Q}(b)\}$ ;  
   **for all**  $a$  **do**  
     **for all**  $a_p = \text{possible (3-)sets of distances in } h_{a,b}$  **do**  
       hypothesise position for  $b$ , based on lateration;  
       **if** *hypothesised position  $\notin \mathbb{R}^3$*  **then**  
         | continue;  
       **end**  
       select all voters  $\{a_v : (a_v \cup a_p) = a, (a_v \cap a_p) = \emptyset\}$ ;  
       voters support hypothesis when connection  $h_{a_v,b}$   
       is consistent with hypothesised position;  
       voters oppose otherwise;  
       **if** *# of supporters > # of opposers* **then**  
         | RANSAC-score hypothesis is # of supporters;  
         | **if** *RANSAC-score is good enough* **then**  
           | break **while**-loop;  
         | **end**  
       **end**  
     **end**  
   **end**  
**end**

**Algorithm 2.** RANSAC.

### Appendix C. Algorithm: Guiding RANSAC

**Data:**  $c, h_{i,j}, \mathcal{V}$

**Result:** sorted  $c'$  and cliques  $\mathcal{Q}(c)$

**for every  $b$  in  $c$  do**

select all nodes  $a \in \mathcal{V}$  with existing  $h_{a,b}$ ;

create graph  $\mathcal{G}'(b) = (\mathcal{V}(a), \mathcal{E}')$  in which all vertices  $V(a)$  are connected with an edge  $\mathcal{E}'$  if their positions are within  $2r_{\text{comm}}$  of each other;

find cliques  $\mathcal{Q}(b)$  in  $\mathcal{G}'(b)$ ;

**end**

sort  $c$  according to clique size of largest cliques in  $\mathcal{Q}(c)$ ;

**Algorithm 3.** Guiding RANSAC by sorting the candidate nodes.