

---

# Self-Organized Localization in Indoor Environments using the ALF Framework

---

**Juergen Eckert\***

Department of Computer Science, University of Erlangen  
Martensstr. 3, D-91058 Erlangen, Germany  
E-mail: juergen.eckert@informatik.uni-erlangen.de

\*Corresponding author

**Reinhard German**

Department of Computer Science, University of Erlangen  
Martensstr. 3, D-91058 Erlangen, Germany  
E-mail: german@informatik.uni-erlangen.de

**Falko Dressler**

Institute of Computer Science, University of Innsbruck  
Technikerstr. 21a, A-6020 Innsbruck, Austria  
E-mail: falko.dressler@uibk.ac.at

**Abstract:** We present and discuss the Autonomous Localization Framework (ALF), a self-organizing indoor localization environment. Location awareness is an important property for a growing number of applications. GPS is frequently used to provide this information in outdoor environments, but this is not applicable for indoor applications. There have been many approaches to solve the localization problem for those GPS-denied scenarios. However, many of them are limited to certain hardware restrictions or do not provide robust self-localization in dynamic real world application. ALF is a complete and modular framework based on minimal hardware requirements. The system is not only capable to deploying itself autonomously in unknown environments and offering position information among the participants, but it also supports accurate real-time localization to customers. The concepts allows to remove or to add features (e.g., the heading of nodes or certain real-time capabilities) as the scenario demands or the even the used hardware changes. The awareness and handling of measurement errors, especially in Non Line of Sight (NLOS) cases, is an essential criterion for a real world application.

**Keywords:** Accurate indoor localization; Self-organization; Distributed wireless sensor network; Flying robots; Anchor-free; None line of sight

**Reference** to this paper should be made as follows:

Eckert, J., German, R., and Dressler, F. (2013) ‘Self-Organized Localization in Indoor Environments using the ALF Framework’, *International Journal of Communications Networks and Distributed Systems (IJCNDS)*, Vol. 10, No. 2, pp.102–122.

**Biographical notes:** Juergen Eckert was born in Weiden i.d.Opf., Germany, in 1983. He received his B.Sc. and his M.Sc. with distinction in computational engineering from the University of Erlangen, Germany, in 2006 and 2008, respectively. He is currently a Ph.D. candidate under supervision of Professor Reinhard German at the computer networks and communication systems groups at the Dept. of Computer Science. His main research interest is in the fields of robotics and wireless sensor networks.

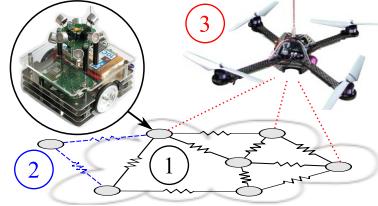
Reinhard German received a diploma in computer science in 1991, a Ph.D. in 1994 and his Habilitation degree in 2000 from the Computer Science Department, Technical University of Berlin. Then he joined the Department of Computer Science at the University Erlangen-Nuremberg first as an associate professor (system simulation) and since 2004 as a full professor (computer networks and communication systems) where he currently serves as dean of the school of engineering. His research interests include performance and dependability analysis of networks, modeling and simulation, model-based testing, numerical analysis of Markovian and non-Markovian models, and applications in vehicular and industrial communications.

Falko Dressler a full professor for computer science at the University of Innsbruck. Before moving to Austria, he was an assistant professor with the Computer Networks and Communication Systems group at the University of Erlangen, Germany. He is an Editor acted as guest editor for journals such as Elsevier Ad Hoc Networks, ACM/Springer Wireless Networks, Elsevier Nano Communication Networks, and IEEE Journal on Selected Areas in Communications. Among others, he wrote the textbook Self-Organization in Sensor and Actor Networks (Wiley, 2007). He is an IEEE Distinguished Lecturer in the fields of inter-vehicular communication, self-organization, and bio-inspired networking. Dr. Dressler is a Senior Member of the IEEE (Communications Society, Computer Society, Vehicular Technology Society) as well as a Senior Member of ACM (SIGMOBILE), and member of GI (KuVS, Real-time). His research activities are focused on self-organizing networks addressing issues in wireless ad hoc and sensor networks, inter-vehicular communication systems, bio-inspired networking, and adaptive network security techniques.

## 1 Introduction

Location awareness is an important topic for mobile systems. In the field of personal computing, location-based services enabled a hitherto unknown type of interacting with an application. In other domains such as Sensor and Actor Networks (SANET), it is at least equally important (Hu and Evans 2004, Lazos and Poovendran 2005). Such intelligent distributed computing systems can, in general, be summarized as *Smarter Planet* solutions. Position information are an essential key feature to leverage new information technology and networks, to create new capabilities, and to provide a more intelligent way to perform common functions.

Typically, all localization approaches have one common characteristic: a reference is needed before a location can be determined. This can be either a



**Figure 1** Conceptual model of our Autonomous Localization Framework (ALF)

map (showing obstacles) or a pre-installed active (or even passive) reference grid consisting of so called landmarks. GPS has become a *de facto* standard for outdoor localization. In GPS-denied areas like indoor environments, or if a higher accuracy is required, other techniques need to be used. Due to of the high variance of system parameters and scenarios, this is still a hot research topic. Besides of hardware issues related to sensing and measurement capabilities, the robust and autonomous establishment of a localization reference grid is one of the most challenging tasks (Eckert, Villanueva, German and Dressler 2011b, Le et al. 2008). In this paper, we show how to set up such a reference system in a fully self-organizing way and how to use this reference system for accurate localization of highly mobile systems such as flying quadrocopters.

The demands in terms of accuracy, cost, power consumption, and others change with the application scenario. Our key objective is to reduce the hardware requirements to a minimum due to cost and energy issues. Therefore, we can not use Simultaneous Localization and Mapping (SLAM) techniques, which are usually based on expensive laser distance sensors and/or resource-intensive image evaluation (Hochdorfer and Schlegel 2010). We started with developing a cheap mobile robot platforms for providing a reference grid to localization customers. These robots are equipped with a short range radio for communication and with an Ultrasound (US) ranging device for neighbor and customer detection. Using a self-organizing approach, no global stages or knowledge can be assumed or will be generated during system runtime (Eckert, German and Dressler 2011b).

Figure 1 depicts our Autonomous Localization Framework (ALF). Only based on inter-node distance measurements, mobile nodes are autonomously deploying themselves on the ground and start forming a robust reference grid. A customer, e.g. a quadrocopter, can use this system to determine its location. A fully distributed and self-organizing algorithm creates and maintains the reference grid using an enhanced Mass-Spring-Relaxation (MSR) technique (step 1 in Figure 1). An initial localization of potential grid nodes helps to prevent the system from oscillating (step 2). The same technique is also used for customer localization (step 3) (Eckert, German and Dressler 2011b).

Flying systems such as quadrocopters pose hard real-time requirements on the localization system. Our Autonomous Localization Framework has been designed especially for this application. We believe that the same system is well suitable for any self-localization approach that is based on autonomous mobile systems forming a reference grid such as the system described by Reich et al. (2011). The modular concept always allows it to replace, to add, or to remove individual components as the requirements or hardware abilities change.

The key contribution of this paper, which extends our earlier work in (Eckert, German and Dressler 2011a) can be summarized as follows: We developed a robust and decentralized framework for enabling anchor-free localization support using low-cost robot systems (Section 3). In an experimental setup, we evaluated the localization performance of our system in terms of accuracy and real-time (Section 4). We specifically show that no global state is necessary. We further show that the implementation of the framework is capable of handling challenging *real-world* problems like measurement errors or Non Line of Sight (NLOS). In addition to our preliminary work in (Eckert, German and Dressler 2011a), a more complete overview of the individual core components as well as their interaction is given. New measurements were conducted to explicitly show the real-time capability for increasing network sizes.

## 2 Related Work

In this section, we briefly discuss related approaches to localization, the use of a grid of mobile robots, and NLOS issues.

### 2.1 Localization Systems

The Cricket system (Priyantha 2005) is probably the best known low-cost localization system for indoor usage. It is a derivate from the Active Bat localization system (Harter et al. 2002) and relies, like its predecessor, on US-based Time of Flight (ToF) measurements to compute positions. The reported resulting absolute position error is less than 30 cm. Although its design is fully decentralized, a non-negligible effort is required for the reference node deployment and the bootstrapping process for the coordinate system.

Chintalapudi et al. (2010) presented EZ, which is a WiFi-based localization approach. Basically, EZ can be seen as a follow-up to the RADAR system by Bahl and Padmanabhan (2000). Both systems estimate their positions using Received Signal Strength Indicator (RSSI) measurements. Thus, the systems are at least one order of magnitude less accurate compared to those relying on US and ToF measurements. Depending on the scenario and a probability threshold  $P$ , EZ has an accuracy of 2 m to 7 m (for  $P = 0.5$ ). The main advantage is that no special deployment effort is needed to bootstrap the system.

We are trying to combine the advantages of both approaches, the accuracy of US-based systems and the ease of the deployment of the WiFi systems.

### 2.2 Mobility Support / Mobile Robots

In general, mobility support has been identified as a source for improving the localization accuracy (Priyantha et al. 2005, Sichitiu and Ramadurai 2004). Thus, for accurate and low-effort localization, the individual platforms need to be mobile. At the same time, localization is one of the key requirements for robot coordination. In (multi-)robot localization applications, the individual units are usually equipped with quite resource and energy expensive components. Laser distance scanners and computer vision are commonly used approaches (Hochdorfer and Schlegel 2010,

Tasaki et al. 2010). On the positive side, this enables the use of SLAM techniques. However, the resource requirements are extremely high and not feasible in many scenarios, even though the advanced position sensors are actually only needed during the construction of the reference grid, assuming that customer localization should be performed using less expensive technology.

Casteigts et al. (2010) presented a novel approach for uniform robot deployment. Based on spring and angular forces, regular hexagons are formed. This shape is very robust in terms of measurement errors. However, the authors require stable multi-hop communication and the robots may not converge to a fix physical position if new nodes keep arriving or departing from the system. This is a critical condition in terms of lifetime and the ability to serve as a reference for localization requests.

Most recently, Reich et al. (2011) presented a system of mobile robots spanning an area based on connectivity between the systems. This grid could finally also be used for localization approaches, even though the key objective here is to provide connected coverage to optimize the overall system performance and the network lifetime (Dietrich and Dressler 2009).

Our approach is to rely on cheap and simple mobile platforms for deploying a minimal initial sensor array. Although all of our sensors are mobile, this is not a major issue. For example, Shell and Matari (2004) developed a system where mobile platforms have been used to deploy immobile human audible beacons. The same technique can be used in our approach to further reduce the system costs for the sensor deployment.

### *2.3 Non Line of Sight*

Even if the reference nodes could be placed in an intelligent way, measurement errors cannot be excluded. Depending on the signal propagation characteristics, different sources for erroneous measurements need to be considered. One of the most critical issues is the NLOS case. The Cricket system (Priyantha 2005) circumvented this issue by placing the reference node on the ceiling. Therefore, emitters and receivers are always in Line of Sight (LOS) and nearly no measurement errors due to NLOS are to be expected. However, this restricts the scenario to indoor applications and manual deployment of the nodes is needed.

Due to multi-path propagation of NLOS measurement pairs, the localization errors can get very large. A common approach to determine the LOS connections is to execute multiple measurements in order to statistically reduce the error. Opposed to the traditional approach of a time history of range measurements, Güvenç et al. (2008) only require the amplitude and delay statistics of a channel in order to perform a NLOS detection. Another approach has been presented by Chan et al. (2006). Based on probably incorrect measurements, they identify NLOS conditions by evaluation the distribution function of all possible positions.

Both approaches compute 2D positions. In the this paper, we show that NLOS can be identified without additional statistics if 3D positions are available together with some easy to collect meta information.

### 3 Autonomous Localization Framework

In the following, we introduce our Autonomous Localization Framework (ALF), which is based on a number of components for setting up and updating the reference grid and determining customer positions using US measurements. We stepwise explain the different algorithms needed for an optimized localization behavior. Some of the described techniques are optional techniques, introduced to improve the accuracy or to obtain additional system information. The entire framework has been built in a modular way: individual components can be replaced depending on the potentials of the used hardware. It should be noted that two assumptions must be fulfilled: nodes need to be capable of measuring to direct neighbors and they need to be able to exchange information with those neighbors.

The key objectives of the localization framework are

- the initial, fully self-organizing deployment of an irregular reference grid, i.e. use of autonomous mobile robots driving to appropriate places in the environment that span a coordinate system by cooperatively assigning themselves relative (anchor-free) positions,
- the accurate customer localization based on the reference grid, and
- the continuous update of the localization grid if the environmental conditions change, e.g. due to new obstacles or added / removed robot systems.

#### 3.1 Advanced Mass-Spring-Relaxation

ALF is based on accurate localization as a basis technique. In our previous work, we presented a distributed version of the MSR algorithm, the Advanced Mass-Spring-Relaxation (advMSR) (Eckert, Villanueva, German and Dressler 2011b). Our localization algorithm is based on distance measurements and also, if possible, takes Angle of Arrival (AoA) measurements into account. The advMSR is inherently self-organizing, thus, assuming no global knowledge about the available nodes and topology. Basically, the localization generates tuples  $(\text{north}, \text{east})^T$  for each node, which are used by the ALF system.

The strategy and motivation for cooperative self-localizing algorithms can be summarized as follows: The network can be represented as a connected graph  $G = (V, E)$  with a vertex set  $V$  and edge set  $E$ . A node  $i$  is represented as a vertex  $N_i \in V$  and a *connection* between a node pair  $(i, j)$  is represented as an edge  $(N_i, N_j) \in E$  in the graph  $G$ . A *connection* is usually interpreted as the capability to sense a node (determine distance and / or angle). For generalization, we assume that no node knows its position (i.e., the network is anchor-free). However, for evaluation, a unique but unknown map representation  $p_r : V \rightarrow \mathbb{R}^2$  must exist, which maps all vertices  $N_i$  to their physical positions  $(\text{north}, \text{east})^T$ . This representation is usually called *ground truth*. What the advMSR algorithm in a distributed manner basically does is to construct a map representation  $p_v$  (called *virtual coordinate system*) based on local knowledge. This representation  $p_v$  is not unique but (if successful) fulfills all relative attributes of  $p_r$  within the measurement accuracy (here,  $\|\cdot\|_2$  is the euclidean distance norm):

$$\|p_r(N_a) - p_r(N_b)\|_2 = \|p_v(N_a) - p_v(N_b)\|_2; \forall N_a, \forall N_b \in V \quad (1)$$

Due to the lack of anchors which usually correlate  $p_r$  and  $p_v$ , the systems may be arbitrary shifted, mirrored, and rotated.

We already reported the principles of the nature-inspired position estimation for the representation  $p_v$  and the maintenance process of the reference grid (Eckert et al. 2010, Eckert, Villanueva, German and Dressler 2011b). In the following, we assume the availability of such localization techniques and discuss system oriented aspects of how areas can be explored, how the heading of a node can be determined, as well as communication aspects of the framework. A key focus is on the handling of measurement errors due to NLOS situations.

### 3.2 Area Exploration

In the initialization phase, we assume that all nodes start at a similar location and need to be uniformly distributed over a given but unknown area. In consequence, the nodes need to be able to autonomously determine their initial coordinates (relative to each other) to finally spread over the area.

In the following, first, the localization accuracy and the probability for a successful setup are determined. Basically, the more neighbors are allowed per node and the *better* they are placed the more accurate the system will get. However, a higher neighbor connectivity also means that the costs will increase significantly. Secondly, the total system costs are defined. This is obviously a trade-off between area coverage, accuracy, and cost.

Similar to the MSR-based grid maintenance algorithm, we again used a force vector approach. We define different zones around each node to optimize the node distribution process. Let  $r_{min}$  and  $r_{max}$  define two radii, which divide the area  $Z_i$  around the origin  $p_r(N_i)$  of each node  $N_i$  into three zones (here,  $\|\cdot\|_2$  is the euclidean distance norm). Then, the zones can be defined as:

$$Z_{restricted,i} := \{\vec{x} \in \mathbb{R}^2 \mid 0 < \|p_r(N_i) - \vec{x}\|_2 < r_{min}\} \quad (2)$$

$$Z_{desired,i} := \{\vec{x} \in \mathbb{R}^2 \mid r_{min} \leq \|p_r(N_i) - \vec{x}\|_2 \leq r_{max}\} \quad (3)$$

$$Z_{attractive,i} := \{\vec{x} \in \mathbb{R}^2 \mid r_{max} < \|p_r(N_i) - \vec{x}\|_2 < \infty\} \quad (4)$$

As the zone names indicate, neighboring nodes are not allowed within  $Z_{restricted,i}$ . Thus,  $r_{min}$  indirectly defines the maximum density of nodes and the associated maximum system costs per ground area.  $Z_{desired,i}$  is the desired belt around a node  $N_i$  where neighbors should be located. Node  $N_i$  will only connect to nodes within this region. The number of neighboring nodes within this belt  $|Z_{desired,i}|$  is equal to the edge degree  $d(N_i)$  of node  $N_i$  from a graph theory point of view ( $|Z_{desired,i}| = d(N_i)$ ). A threshold  $\lambda$  is used to control the system costs. This threshold is basically limiting the maximum connectivity degree:  $3 \leq d(N_i) < \lambda$ , where 3 is the minimum number of required connections for calculating a 2D position.

Again, the higher the connectivity  $d(N_i)$  gets, the more accurate localization results can be obtained, but at increasing system costs. This problem has already been investigated in the literature. A typical value reported in the literature is in the range of 8 to 13 connections per node (Priyantha et al. 2003). We experimented with those values in the scope of the ALF framework. In most experiments, already

**Table 1** Placement decision constraints

Guard	Action	Use Nodes ( $Z_c$ )
$ Z_{restricted,i}  > 0$	drive	$Z_i \setminus Z_{desired,i}$
$3 \leq  Z_{desired,i}  < \lambda$	place found	-
$\lambda \leq  Z_{desired,i} $	drive using $\frac{1}{2}\hat{d}_{i,j}$	$Z_{desired,i}$
$ Z_{attractive,i}  > 0$	drive	$Z_{attractive,i}$
else	random drive	-

a connectivity degree of 6 led to astonishingly good results. N.B.:  $\lambda$  should not get below the error correction threshold of 4 (see Subsection 3.6).

Furthermore, the radii  $r_{min}$  and  $r_{max}$  must be chosen according to the optimal detection range of the used distance measurement hardware. The closer both radii get, the more regular distributions of the nodes will be achieved. However, the more difficult it becomes for the algorithm to find a valid solution. Of course,  $r_{max}$  should also not be set close to the maximum of the detection range in which the distance measurement quality decreases in a non-linear way.

During the positioning phase of each node  $N_i$ , distances  $\hat{d}_{i,j}$  to all detectable  $j$  neighbors are measured and sorted into the three introduced sets. The resulting actions to be taken depend on four guards that are evaluated serially:

1. if neighbors are in the restricted zone, the node has to find a more appropriate position;
2. if a sufficient number of nodes are in the desired zone ( $3 \leq |Z_{desired,i}| < \lambda$ ), the node position can be fixed;
3. if too many nodes are in the desired zone, the node will move away from those nodes;
4. if the node has to move and if there are nodes in the attractive zone, the node will move towards those.

In any other case, the node will start searching randomly for better locations. All the actions are summarized in Table 1.

In the following, we briefly introduce the force vector approach for repositioning nodes to form the reference grid. All the calculations are performed locally at a node, only taking the measurements to neighboring systems into account.

Node  $N_i$  is connected to each node from the selected set with a spring of equilibrium length of  $l_0 = \frac{r_{min}+r_{max}}{2}$ . Based on the chosen set  $Z_c$  ( $|Z_c| = j$ ), the force vector  $\vec{F}_i$  for the movement is computed according the basic MSR as depicted in Equation 5, where  $k_{i,j}$  is the spring constant (usually set to 1),  $\vec{e}_{i,j}$  characterizes the unit vector from node  $N_i$  to node  $N_j \in Z_c$ , and  $\hat{d}_{i,j}$  represents the corresponding measured distance.

$$\vec{F}_i = \sum_{N_j \in Z_c} \vec{F}_{i,j} = \sum_{N_j \in Z_c} -\vec{e}_{i,j} k_{i,j} (\hat{d}_{i,j} - l_0) \quad (5)$$

The unit vector  $\vec{e}_{i,j}$  can be obtained using two different approaches. Assuming all nodes in  $Z_c$  know their positions, thus, node  $N_i$  can localize itself according

to these references and finally apply simple geometrical calculations. The problem is that parallel operation and bootstrapping become extremely difficult. Another approach is to measure angle information. Most distance ranging devices are based on multiple sensors; our system uses four independent US sensors (Eckert, German and Dressler 2011b). Thus, the AoA can be estimated. Our system provides an accuracy of at least  $\pm 45^\circ$ , which is sufficient for the computations.

The final driving direction and distance are proportional to the force vector  $\vec{F}_i$ . Small measurement uncertainties and the mobility of nodes helps to prevent oscillations in the system to achieve fast convergence. N.B.: this scheme works well both in 2D as well as in 3D.

### 3.3 Heading

The previously mentioned AoA estimation can also help to estimate the heading of the platform, which is a necessary measure in various scenarios. In theory, the heading  $\Psi_i$  of a node  $N_i$  can be computed using one arbitrary neighbor  $N_j$ . According to the positions  $p_v(N_i)$  and  $p_v(N_j)$ , the absolute heading  $\Psi_{i,j}^{north} = atan2(p_v(N_j) - p_v(N_i))$  in relation to the north pole (which is the commonly assumed as  $0^\circ$ ) can be determined. Subtracting the measured AoA  $\hat{\Psi}_{i,j}$  results in the true heading  $\Psi_{i,j} = \Psi_{i,j}^{north} - \hat{\Psi}_{i,j}$ .

The described calculation becomes error prone in real scenarios due to biased measurements and positioning errors. Simple averaging over all  $j$  headings  $\Psi_{i,j}$  is not possible because of two reasons. First, there is a non-linearity due to the overflow of the co-domain ( $\Psi_{i,j} \in (-\pi, +\pi]$ ), which might lead to wrong average values. Secondly,  $p_v(N_i)$ ,  $p_v(N_j)$  or  $\hat{\Psi}_{i,j}$  might be error prone and heavily affect the results.

The overflow of the domain can be compensated by moving from angles to a vector-based representation:

$$\vec{\psi}_{i,j} = (\sin(\Psi_{i,j}), \cos(\Psi_{i,j}))^T \quad (6)$$

However, the second issue is more challenging. Summing up all heading vectors already allows to conclude to a fairly good estimation. This can further be improved by adding a weight indicating the confidence to each vector. For our distance measurement hardware, we see that the shorter the measured distance is, the more accurate the measured AoA gets (Eckert, German and Dressler 2011b). Thus, the weighting function  $\kappa(\cdot)$  returns larger values for nodes in close proximity.

The overall heading can be computed described in Equation 7. The angle  $\Psi_i = atan2(\vec{\psi}_i)$  of the resulting vector  $\vec{\psi}_i$  represents the heading of node  $N_i$ . The length of the vector  $\|\vec{\psi}_i\|_2$  allows to conclude the confidence of this information. It is normalized to  $\kappa(\cdot)$ .

$$\vec{\psi}_i = \frac{1}{j} \sum_{N_j} \kappa(N_j) \cdot \vec{\psi}_{i,j} \quad (7)$$

### 3.4 Communication

For the ALF system, no specific communication topology is required. A sufficient criterion is that each node  $N_i$  must be able to communicate with all of its  $j$

neighboring nodes  $N_j \in \hat{N}_i := \{N_n \in V \mid \|p_r(N_i) - p_r(N_n)\|_2 \leq r_{max}\}$  (where  $V$  represents the set of nodes). Multicast communication to all neighbors would be the optimal solution for this application, because most of the transferred information needs to be made available to all neighbors simultaneously. In wireless networks, broadcast is an appropriate alternative.

In order to join an existing network, a node  $N_i$  registers to all of its neighbors  $N_j$  after finding an adequate position. This registration needs to be reliable. Thus, reliable unicast is needed for this process. If new or updated information (e.g., position  $p_v(N_i)$ , neighbor  $N_j$ , measurement  $\hat{d}_{i,j}$ ) becomes available at node  $N_i$ , it pushes this data to all of its neighbors. These in turn evaluate the data and perform the necessary actions. This data will be transmitted more frequently. Therefore, no reliable communication channel is required.

As a result, the amount of transmitted data packets is reduced to a minimum and it also provides mechanisms for autonomously entering energy saving standby states after system convergence has been reached. Also, no network-wide flooding is needed.

### 3.5 Customer Localization

ALF is capable of cooperatively assigning positions  $p_v(N_i)$  and headings  $\Psi_i$  to network participants  $N_i$ . The used advMSR algorithm is quite accurate but may take a relatively long time to reach convergence. The network gets stabilized as long as involved nodes can stay within a given area, waiting for advMSR completion, and will stay there for a long time. In contrast to this, there may exist nodes  $N_c$ , which are not able to remain at a position (e.g., flying objects) or which should not participate in the network  $V$  (e.g., due to lifetime issues). However, even these nodes can benefit from the localization service in the network. We call this *customer localization*.

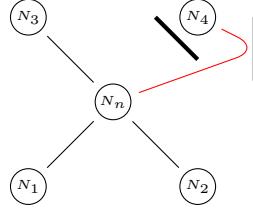
The following properties need to be satisfied:

- customers  $N_c$  can be localized according to the reference grid,
- the resulting position  $p_v(N_c) \in \mathbb{R}^3$  is provided in time, and
- the reference grid  $V$  is not influenced by customers  $N_c \notin V$ .

In our previous work, we presented a framework, which is specifically designed for this purpose (Eckert, German and Dressler 2011b). The strategy can be summarized as follows. A mobile node  $N_c \notin V$  triggers a distance measurement. The resulting data is distributed over the reference nodes  $N_i \in V$  in range. Subsequently node  $N_c$  emits an agent (radio package), which collects the distance measurements  $\hat{d}_{c,i}$ . Within a hard deadline the broadcast-based application layer protocol returns as many measurement tuples  $\mu(N_i) := (\hat{d}_{c,i}, p_v(N_i))^T$  as possible to the initiator, which, in turn, evaluates these tuples according to time and entropy concerns. Exploiting this gathered information the position  $p_v(N_c) \in \mathbb{R}^3$  of the customer  $N_c$  can be computed.

### 3.6 Measurement Errors

Usually, it is assumed that a measurement path is symmetric, i.e.  $\hat{d}_{i,j} = \hat{d}_{j,i}$ . However, this only holds in approx. 90 % of our measurements and, depending on



**Figure 2** NLOS scenario (top view)

the scenario, may be even worse. A bad alignment between emitter and receiver of the US distance measurements can falsify the result depending on the direction, i.e. not a direct but a reflected signal triggers the measurement. Those incorrect measurements cannot be distinguished from correct measurements as the jitter is equal (the worst case scenario is static without mobility).

We try to correct such errors using a rather simple error model: Both directions need to be measured to identify asymmetric measurements. As we use ToF, the shortest flight time must be the most correct measurement ( $\tilde{d}_{i,j}$  represents the measurement from node  $N_i$  to node  $N_j$ ):

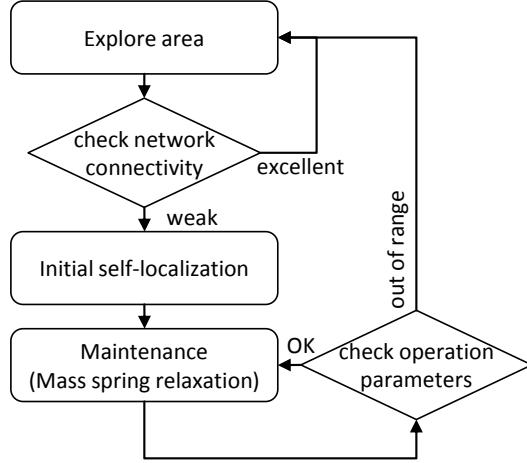
$$\hat{d}_{i,j} = \hat{d}_{j,i} = \min(\tilde{d}_{i,j}, \tilde{d}_{j,i}) \quad (8)$$

If  $\epsilon_{i,j} = |\tilde{d}_{i,j} - \tilde{d}_{j,i}| > \theta$ , where  $\theta$  represents the hardware specific error range, then the node that measured the larger distance can also not rely on its AoA estimation because it received the reflected signal. Overall, this solution gave us almost 100 % reliability for the LOS cases.

Detecting NLOS links is even more challenging. Consider the example depicted in Figure 2. This network consists of four initially perfectly distributed nodes  $N_1$ ,  $N_2$ ,  $N_3$ , and  $N_4$  at positions  $(0, 0)$ ,  $(0, 1)$ ,  $(1, 0)$ , and  $(1, 1)$ . The newly arriving node  $N_n$  starts the localization procedure to subsequently join the network. Node  $N_n$  has a NLOS link to node  $N_4$ , all other links are LOS. None of the participating nodes can distinguish between LOS and NLOS.

The distances will be determined as  $\hat{d}_{n,i} = 2^{-\frac{1}{2}}; i \in [1, 3]; \hat{d}_{n,4} = 1$ . Thus, using basic trilateration techniques (Eckert, German and Dressler 2011b), the node will be located at four equally weighted positions:  $(0.5, 0.5)$ ,  $(0.25, 0.25)$ ,  $(0.25, 0.5)$ , and  $(0.5, 0.25)$ . From this 2D view it is neither possible to determine the correct position nor to identify the outliers.

However, from a 3D point of view and by adding additional knowledge about the height of a system (in our case, all the robots are driving on the floor), the correct position can be obtained by looking at the Z-coordinates:  $0.0$ ,  $\text{NaN}$ ,  $0.43$ , and  $0.43$ . Obviously, the first coordinate  $(0.5, 0.5, 0.0)$  must be the correct result. Measurements that result in wrong Z-coordinates are considered NLOS links and need to be blacklisted in the localization framework. Without any loss of generality, this approach can be applied to almost any localization scheme. Obviously, additional information is necessary to identify NLOS cases. For best results, additional hardware components are needed to minimize such failures. For obtaining height information, for example simple pressure sensors could be used providing a relative accuracy of a few centimeters.



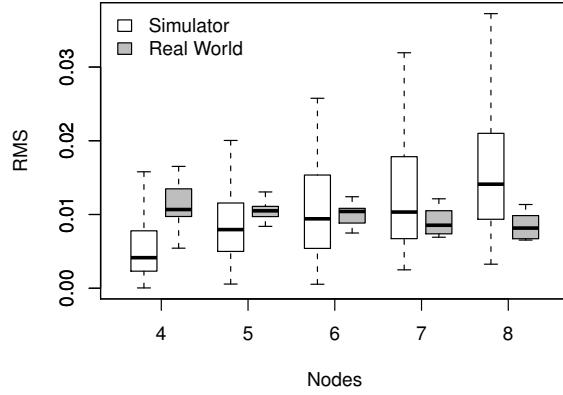
**Figure 3** System behavior of the ALF localization framework

In general, NLOS can be detected if three measurement tuples are available together with additional height information. If one or more additional tuples are available all NLOS links can be identified as long as three correct LOS links are available.

The third important source of error is the co-called *flip ambiguity* problem, which is typical for almost any localization technique (Kannan et al. 2010). Due to small measurement errors, nodes may get wrong initial coordinates. This typically happens if the reference nodes for the localization are nearly collinear placed. In such situations, very small measurement errors significantly falsify the Z-component. Thus, it is very likely that those get always ignored. We counteract this issue exploiting the mobility of our nodes, using short movements in random directions.

### 3.7 Network Bootstrapping

An initial network  $V'$  needs to be constructed before nodes can regularly join. Three nodes ( $|V'| = 3$ ) are sufficient to span up a plain. However, to avoid placement errors at the very beginning, a fourth node is required. All four nodes need to be fully connected and well placed to safely bootstrap the grid: Three nodes generate the coordinate system and the fourth node localizes itself according to this trio to verify that there is no NLOS in the initial network (see Section 3.6). In order to speed up the initial bootstrapping, we implemented the bootstrapping in a central manner, even though the localization process is fully decentralized. A root node requests all measurements from three nodes, performs all the computations related to the initial localization step, and, afterwards, assigns the initial positions. This central approach with very small computational efforts clearly outperforms other solutions that require clearly more time for the initialization and a high number of transmissions to identify NLOS issues.



**Figure 4** Validation of simulation and experimental results

### 3.8 Overall System Behavior

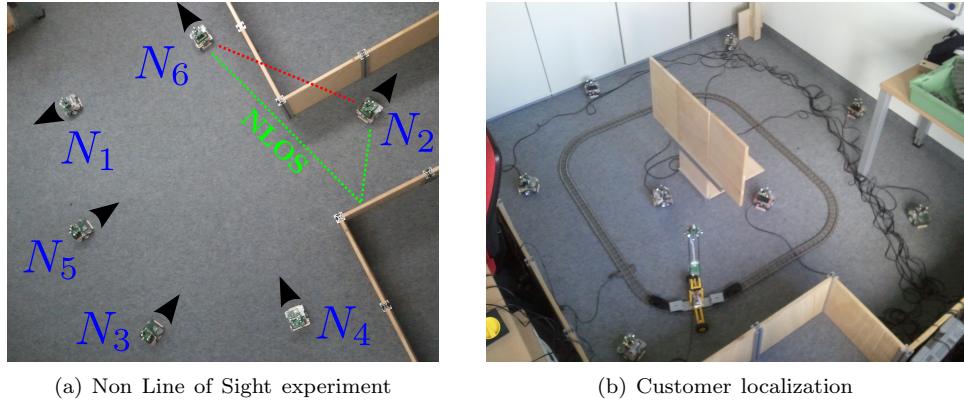
The system behavior of ALF for each node  $N_i$  is summarized in Figure 3. After starting the system, each node is searching for an existing localization grid. If the particular node is immobile, it is also feasible to sleep for a certain time and then retry to establish a connection. As soon as the node finds a sparsely covered part of the network, it starts the initial self-localization (Eckert, German and Dressler 2011b) before it can become part of the grid by registering with the neighbors in the grid. The robot then continuously tries to improve its initial position using our advMSR technique. As soon as the estimated localization error falls below a certain threshold, the robot is ready to serve customers' localization requests. In theory, the system can stay in this state for a very long time. No additional measurements, computations, or transmissions need to be performed. However, if errors can not be solved or if the connectivity level falls below a predefined threshold, the robot can re-enter the initial phase and move to another position. No global knowledge or additional synchronization is involved to establish the grid.

## 4 Evaluation

In order to evaluate the localization performance of the Autonomous Localization Framework, we performed a number of experiments, each focusing on specific characteristics of the algorithm. First, we briefly show that the used experimental setup, i.e. our localization hardware, performs the advMSR algorithm with similar accuracy as previously estimated in simulations (Eckert, Villanueva, German and Dressler 2011b). We then carefully evaluated the behavior in NLOS situations, before finally assessing the overall system performance.

### 4.1 Advanced Mass-Spring-Relaxation Performance

The core of the ALF framework is based on the Advanced Mass-Spring-Relaxation technique. It has been carefully examined in a custom build simulator for network sizes of up to 1000 nodes (Eckert, Villanueva, German and Dressler 2011b). The



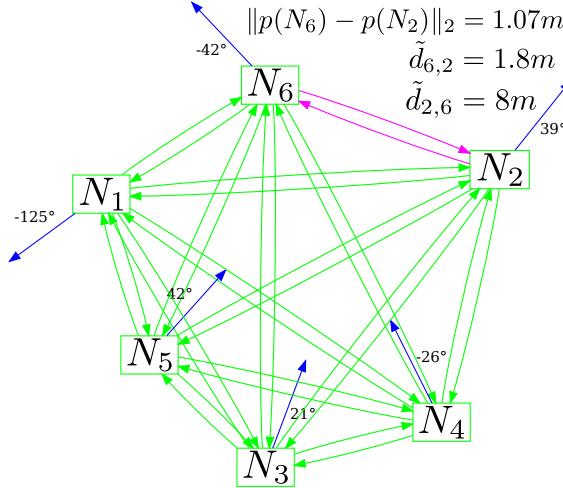
**Figure 5** Real world experimental setup

algorithm only interacts with direct neighbors and does not require any global knowledge. The advMSR self-localization performed extremely well in simulation as previously reported. In order to validate the algorithm before starting to assess the overall performance of our framework, we repeated those simulation experiments with lower node numbers and also executed similar experiments using our localization hardware. Basically, up to 8 nodes were placed randomly in the environment before starting the advMSR procedure. Figure 4 shows that for real world experiments the constructed network representation accuracy is within the co-domain of our simulator. The evaluation is based on relative node distances between the calculated positions and the actual ones. The Root Mean Square (RMS) of the normalized error of both the simulation and the lab experiments are similar. The RMS has been identified as an accurate measure to compare self-localization solutions (Eckert, Villanueva, German and Dressler 2011a). The figure depicts the RMS for different network sizes in form of boxplots. The thick line represents the median. The rectangular boxes contains 50 % of the measurements indicating the 25 % and 75 % quantiles. Finally, the whiskers show the 2.5 % and 97.5 % quantiles.

#### 4.2 NLOS

Figure 5(a) shows a typical NLOS situation. The edge between nodes  $N_6$  and  $N_2$  cannot be measured correctly due to an obstacle. Multipath propagation effects result in the following measured distances:  $\tilde{d}_{6,2} = 1.8 \text{ m}$ ,  $\tilde{d}_{2,6} = 8 \text{ m}$ , whereas the true physical distance is  $d_{6,2} = 1.04 \text{ m}$ . The accumulated results as used internally by ALF of a selected run are depicted in Figure 6. In this figure, all the nodes, their headings, as well as the connecting links are drawn. Green links are correct from the algorithm's point of view. Magenta links indicate blacklisted ones, which are not considered for further computations. As can be seen, ALF reliable detected and ignored only the NLOS link. The distance error was  $||p_v(N_6) - p_v(N_2)||_2 - d_{6,2}| = 3 \text{ cm}$ . Without this blacklisting scheme, no correct solution could be found.

In total, we conducted two experiments (ten repetitions for each) with the same setup. For the NLOS detection, a threshold  $\xi_A$  is needed to decide whether to accept the computed position or not. For the first experiment, we used  $\xi_A =$



**Figure 6** Resulting network representation

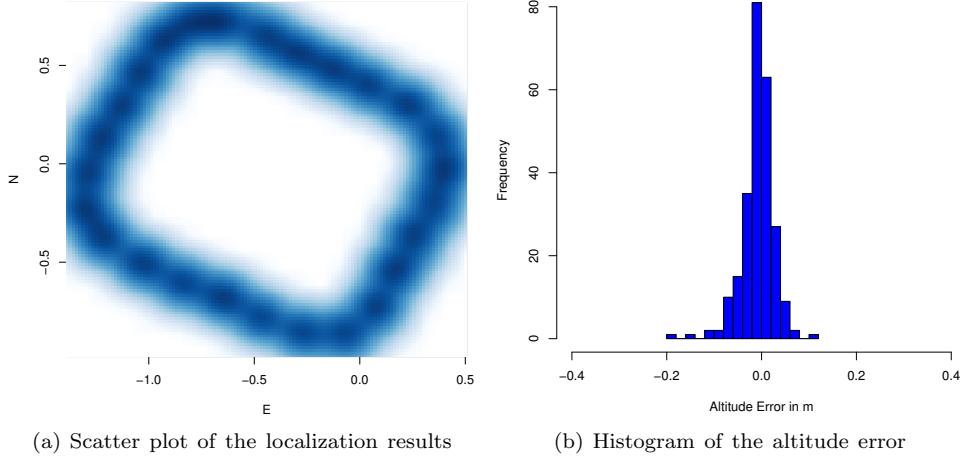
$6\theta$ , where  $\theta$  represents the hardware specific error range. In three out of ten repetitions, it was not possible to correctly detect the NLOS link during the first initialization. However, after some nodes autonomously disconnected and reconnected (see Figure 3 *parameters out of range*), a valid solution could be found. For the second set of experiments, we used a smaller threshold of  $\xi_A = 4\theta$ . In this configuration, the system detected the NLOS link with 100 % accuracy.

### 4.3 Overall System Performance

ALF has primarily been designed to provide self-localization services to mobile customers such as flying quadrocopters. Unfortunately, it is not easily possible to accurately steer a quadrocopter on a given trajectory to verify the localization system's accuracy. Thus, we used a toy train to simulate the quadrocopter by moving the localization unit on a well defined trajectory. In our experiment, the sensing device was mounted on a stick at an altitude of 64 cm. It is driving on a rectangularly shaped trajectory through an autonomously constructed reference grid consisting of nine nodes. In this setup, straightforward and curve movement can be investigated. Figure 5(b) shows a picture of the testbed. For NLOS measurements, we placed an obstacle in the middle of the set. This results in at least  $P_{mf} = 30\%$  corrupted measurements. Similar values have been reported in the literature, e.g. by Casas et al. (2006).

#### 4.3.1 LOS

In a first experiment and as an reference, we removed the obstacle in the middle of the set. Figure 7(a) depicts the resulting coordinates for the client in form of a scatter plot for the top view. The train was driving 10 times around the railroad line with a speed of approx.  $1 \text{ m s}^{-1}$ . The darker an area gets the more often the sensor was detected in this area. The rectangle of the railroad can clearly be identified. As the mechanical construction on the train was slightly unstable, the sensing hardware

**Figure 7** Line of Sight experiment**Table 2** Altitude error in meter

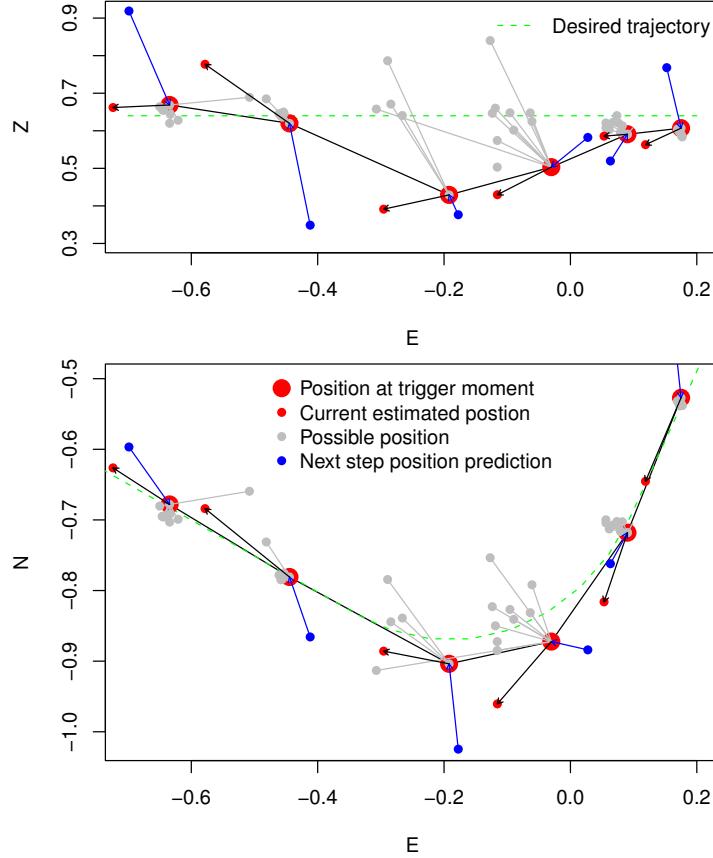
Exp.	Min	1 <sup>st</sup> Qu.	Median	Mean	3 <sup>rd</sup> Qu.	Max
LOS	-0.191	-0.022	-0.005	-0.008	0.011	0.111
NLOS	-0.377	-0.025	-0.002	-0.009	0.020	0.348

can swing a few centimeters on the top of the stick. These oscillations distort the integrated prediction filter, resulting in incorrect positions. In contrast to our initial approach (Eckert, German and Dressler 2011b), we used a linear multistep method (Adams-Basforth-Method) to improve the predication in the curves.

Measuring the actual position of the train on the ground is unfortunately too inaccurate, because we had no odometry available with an accumulated error smaller than a few centimeter. Therefore, we evaluated the measured altitude. The altitude of the sensing device was exactly constant (64 cm; the altitude change due to the oscillations is negligible). A histogram for the altitude error is shown in Figure 7(b). As can be seen, the accuracy is very high. Numerical values presented in Table 2 indicate that 50 % of the measurements are within  $\pm 2.2$  cm of the correct altitude. The maximum measurement error was 19.1 cm.

Figure 8 shows individual measurements for a small part of the railroad. The upper plot depicts the altitude measures and estimations for different East-coordinates. The lower plots shows North-East-coordinates. In our experiment, the train drove counterclockwise. The original railroad as well as the fixed altitude are drawn in dashed green lines. In our plots, the big red dots depict accepted sampling points for the client localization.

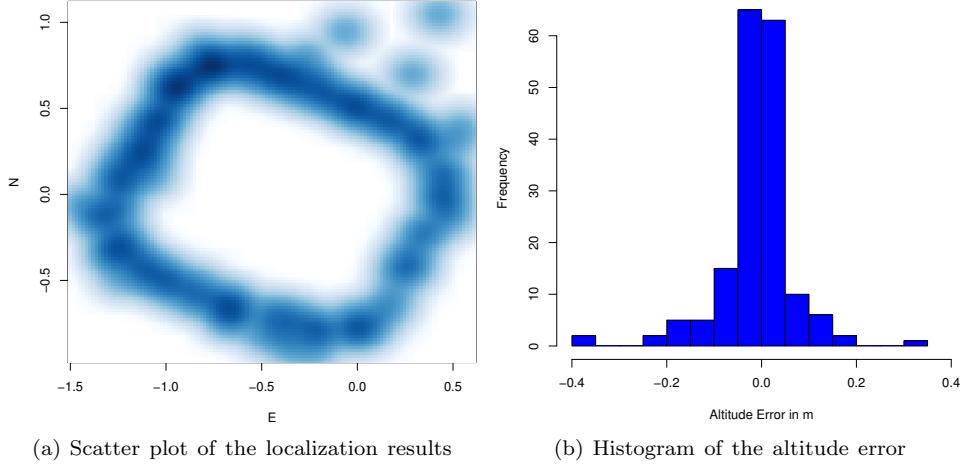
We collected all the measurement information every 200 ms. Between these intervals, the train moves continuously. As soon as the latest sampling position is known to the system, the current position is estimated (represented in form of small red dots in Figure 8) based on the last sampling points and the elapsed time. This process works as follows: Before a sampling point is accepted, it is compared to



**Figure 8** Trajectory of the localization experiment

the predicted position using the last sampling points (small blue dots in Figure 8). The best matching position out of all current measurements (small gray dots in Figure 8) is chosen as the new sampling point. More details on the estimation can be found in (Eckert, German and Dressler 2011b).

In the plot, it can easily be seen that both the prediction and the estimation process have certain inertia as they rely on the same sampling points. Therefore, as the train moves around the corner, the outer measurements get chosen as sampling points (Figure 8, lower plot). This represents the worst case scenario in our experiment. We picked this example to show the following problem: In this situation, the altitude of the possible positions (grey) drops with the distance to the center of the railroad due to the probe reaches the border of the network as well as due to measurement errors (Figure 8, upper plot). We furthermore see that more suitable positions are available but never chosen according to the prediction model. Using clients such as typical quadrocopter that provide internal sensors (e.g., gyroscopes and accelerometers), this information can be used to improve the prediction and to avoid those measurement errors.

**Figure 9** Non Line of Sight experiment

#### 4.3.2 NLOS

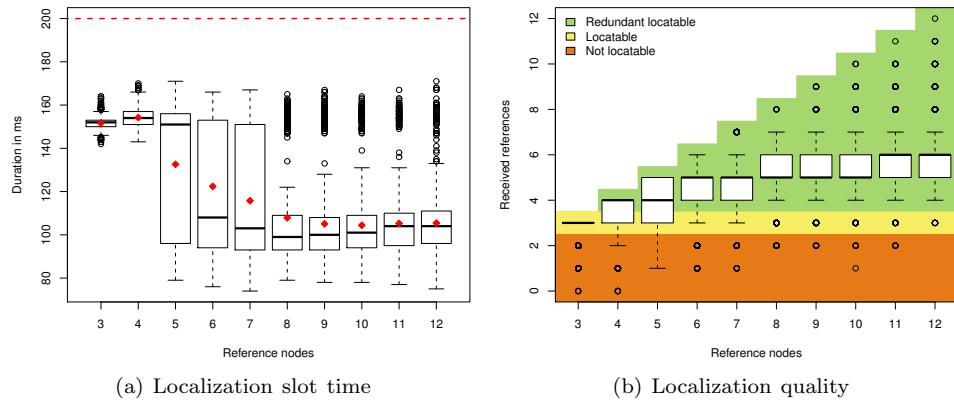
Figure 9(a) depicts the resulting coordinates of the NLOS scenario. We used the same conditions as described for the LOS measurements, except that we used an obstacle in the middle of the system. In contrast to the LOS case, more noise can be observed but the rectangle can still clearly be identified. However, four positions can be perceived outside of the trajectory, which represent outliers in the localization experiment. In 2 out of 10 rounds, the position information got corrupted (actually, always in the same curve) due to significant oscillations ( $\pm 8$  cm) in combination with NLOS.

A histogram showing the altitude error (Figure 9(b)) allows to assess the localization accuracy. The numerical values in Table 2 indicate that 50 % of the measurements have an error of less than  $\pm 2.5$  cm. The significant outliers of nearly  $\pm 40$  cm are significant but happen statistically very infrequently.

#### 4.3.3 Real-time Capabilities

Besides accuracy, timeliness is a mandatory requirement for our chosen scenario. Late localization messages may result in a crash of the flying robot. In our previous work, we introduced a mathematical procedure capable of handling this issue (Eckert, German and Dressler 2011b). An upper bound can be specified for the latest possible time at which a position  $p_v(N_c)$  can robustly be computed out of an arbitrary amount of available measurement tuples  $\mu_c := \{\mu(N_i) | \forall N_i \in \hat{N}_c\}$ . In this work, we also introduced an application layer protocol, which is responsible for collecting the distributed tuples  $\mu_c$ . However, this was only experimentally evaluated with a very limited number of nodes. In the following experiment, we now show that the duration of the localization procedure does not increase with the number of neighboring reference nodes in range  $|\hat{N}_c|$ . N.B.:  $|\hat{N}_c|$  is not the total network size  $|V|$  but the connectivity degree  $d(N_c)$  of the customer  $N_c$  ( $|\hat{N}_c| = d(N_c) \leq |V|$ ).

The participating grid nodes  $\forall N_i \in \hat{N}_c$  have no reliable knowledge on how many tuples have been received already by customer  $N_c$  and on how many other nodes



**Figure 10** Real-time experiments

try to send their information to the customer. In general, it holds that the wireless channel must not be overloaded; otherwise the information will be received too late.

For this experiment, we arranged  $|\hat{N}_c| \in [3, 12]$  reference nodes around one customer  $N_c$  in close proximity to stepwise increase the network load. Figure 10(a) shows the total time (tuple collection and subsequent computation) for the position estimation for different reference grid sizes  $|\hat{N}_c|$ . Again, we used boxplots to depict the results. The hard deadline is marked with the red dashed line (as stated before, 200 ms is our measurement rate). Each localization attempt must be completed within this time. Additionally, the red rhombi represent the average duration and the black circles indicate statistical outliers. The plot summarizes 13 500 measurements. It can be seen that not a single attempt passed the threshold even for increasing connectivity degrees.

The measurement time declines for neighbor counts larger than five. The collection of tuples  $\mu_c$  has to be well balanced between the number of collected tuples  $|\mu_c|$  and, therefore, the consumed time. In general, the faster a reliable amount of tuples can be collected the better. After triggering a ToF measurement, a customer  $N_c$  requests the corresponding tuples from the reference grid and waits. It continues working if a certain amount of tuples  $|\mu_c| \geq \omega$  is available or a deadline has expired. A group of 3 different tuples  $\mu(\cdot)$  builds a subset. It is the minimal requirement for defining the position  $p_v(N_c) \in \mathbb{R}^3$  of customer  $N_c$ . If we assume that we have 5 tuples and we again use the typical value reported in the literature of an undetectable (possibly NLOS) measurement failure of  $P_{mf} = 30\%$ , we can conclude that at least one correct subset, which consists of 3 correct tuples, can be generated from this set with probability  $P_g > 98.5\%$  (Eckert, German and Dressler 2011b). Therefore, the algorithm waits until at least  $\omega = 5$  tuples have arrived.

Figure 10(b) depicts the number of received tuples  $|\mu_c|$  for different node sizes in range  $|\hat{N}_c|$ . The median is saturated ( $P(|\mu_c| \geq 5) \geq 50\%$ ) if at least 6 reference nodes are available. This supports the decrease of the estimation duration. We further colorized three areas of the graph. If 0 to 2 tuples have been received, no position estimation is possible. 3 tuples allow to conclude to exactly one position  $p_v(N_c) \in \mathbb{R}^3$ , which might be corrupted. If at least 4 tuples are available, a redundant localization is possible. A very high accuracy even for 3 or 4 tuples can

**Table 3** Probability distribution (in percent) of received measurement tuples

$ \hat{N}_c $	3	4	5	6	7	8	9	10	11	12
$P( \mu_c  \leq 2)$	18.3	16.3	6.7	5.7	3.5	0.7	0.5	0.3	0.3	0.0
$P( \mu_c  = 3)$	81.7	30.9	22.2	16.2	8.7	5.3	2.5	1.3	0.8	0.6
$P( \mu_c  \geq 4)$	0.0	5.2	71.1	78.1	87.8	94.1	97.1	98.3	98.9	99.4

be achieved using an internal movement model at the customer. If only incorrect measurements are presented by the grid (for a short amount of time), they all will be dropped. Table 3 depicts the percentage of how many reference tuples  $|\mu_c|$  were received for different numbers of grid nodes in range.

According to this experiment, an average connectivity of 8 reference nodes is optimal in terms of cost, duration, and availability. If availability becomes less important (e.g., a customer can wait for a position), then this number can be further decreased.

## 5 Conclusion

We presented a self-organizing, fully decentralized and stateless localization framework, which is capable of autonomously deploying and spanning up a reference grid in unknown environments. Based on this grid, customers such as quadrocopters can be accurately localized in real-time. The modular approach makes it easy to adopt the framework for different abilities and requirements. Additional information such as anchor nodes  $N_a$  ( $p_v(N_a) \equiv p_r(N_a)$ ), which know their physical position can easily be incorporated. Using the MSR theory, we were able to design a system that does not need any *a priori* knowledge or a global database. Therefore, it can easily be used for embedded systems with limited energy and memory resources. In summary, it can be said that our Autonomous Localization Framework is providing very accurate real-time localization. This also holds for handling NLOS situations, which is a strong requirement for real world applications.

## References

- Bahl, P. and Padmanabhan, V. N.: 2000, RADAR: An In-Building RF-based User Location and Tracking System, *19th IEEE Conference on Computer Communications (IEEE INFOCOM 2000)*, Tel-Aviv, Israel.
- Casas, R., Marco, A., Guerrero, J. and Falco, J.: 2006, Robust Estimator for Non-Line-of-Sight Error Mitigation in Indoor Localization, *EURASIP Journal on Applied Signal Processing* **2006**, 1–8.
- Casteigts, A., Albert, J., Chaumette, S., Nayak, A. and Stojmenovic, I.: 2010, Biconnecting a Network of Mobile Robots Using Virtual Angular Forces, *72nd IEEE Vehicular Technology Conference Fall (VTC 2010-Fall)*, IEEE, Ottawa, Canada, pp. 1–5.

- Chan, Y. T., Tsui, W. Y., So, H. C. and Ching, P. C.: 2006, Time-of-Arrival Based Localization Under NLOS Conditions, *IEEE Transactions on Vehicular Technology* **55**(1), 17–24.
- Chintalapudi, K., Iyer, A. P. and Padmanabhan, V. N.: 2010, Indoor Localization Without the Pain, *16th ACM International Conference on Mobile Computing and Networking (MobiCom 2010)*, Chicago, IL, pp. 173–184.
- Dietrich, I. and Dressler, F.: 2009, On the Lifetime of Wireless Sensor Networks, *ACM Transactions on Sensor Networks (TOSN)* **5**(1), 1–39.
- Eckert, J., German, R. and Dressler, F.: 2011a, ALF: An Autonomous Localization Framework for Self-Localization in Indoor Environments, *7th IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS 2011)*, IEEE, Barcelona, Spain.
- Eckert, J., German, R. and Dressler, F.: 2011b, An Indoor Localization Framework for Four-rotor Flying Robots Using Low-power Sensor Nodes, *IEEE Transactions on Instrumentation and Measurement* **60**(2), 336–344.
- Eckert, J., Villanueva, F., German, R. and Dressler, F.: 2010, A Self-Organizing Localization Reference Grid, *ACM SIGMOBILE Mobile Computing and Communications Review* **14**(3), 4–6.
- Eckert, J., Villanueva, F., German, R. and Dressler, F.: 2011a, Considerations on Quality Metrics for Self-localization Algorithms, *5th IEEE/IFIP International Workshop on Self-Organizing Systems (IWSOS 2011)*, Vol. LNCS 6557, Springer, Karlsruhe, Germany, pp. 104–115.
- Eckert, J., Villanueva, F., German, R. and Dressler, F.: 2011b, Distributed Mass-Spring-Relaxation for Anchor-free Self-localization in Sensor and Actor Networks, *20th International Conference on Computer Communication Networks (ICCCN 2011)*, IEEE, Maui, HI.
- Güvenç, I., Chong, C.-C., Watanabe, F. and Inamura, H.: 2008, NLOS identification and weighted least-squares localization for UWB systems using multipath channel statistics, *EURASIP Journal on Advances in Signal Processing* **2008**, 36.
- Harter, A., Hopper, A., Steggles, P., Ward, A. and Webster, P.: 2002, The Anatomy of a Context-Aware Application, *ACM/Springer Wireless Networks* **8**, 187–197.
- Hochdorfer, S. and Schlegel, C.: 2010, 6 DoF SLAM using a ToF Camera: The Challenge of a Continuously Growing Number of Landmarks, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, IEEE, Taipei, Twaiwan, pp. 3981–3986.
- Hu, L. and Evans, D.: 2004, Localization for Mobile Sensor Networks, *10th ACM International Conference on Mobile Computing and Networking (MobiCom 2004)*, ACM, Philadelphia, PA, pp. 45–57.
- Kannan, A. A., Fidan, B. and Mao, G.: 2010, Analysis of Flip Ambiguities for Robust Sensor Network Localization, *IEEE Transactions on Vehicular Technology* **59**(4), 2057–2070.

- Lazos, L. and Poovendran, R.: 2005, SeRLoc: Robust localization for wireless sensor networks, *ACM Transactions on Sensor Networks (TOSN)* **1**(1), 73–100.
- Le, V.-D., Dang, V.-H., Lee, S. and Lee, S.-H.: 2008, Distributed localization in wireless sensor networks based on force-vectors, *International Conference on Intelligent Sensors, Sensor Networks and Information (ISSNIP 2008)*, IEEE, Sydney, Australia, pp. 31–36.
- Priyantha, N. B.: 2005, *The Cricket Indoor Location System*, Phd thesis, Massachusetts Institute of Technology.
- Priyantha, N. B., Balakrishnan, H., Demaine, E. D. and Teller, S.: 2005, Mobile-Assisted Localization in Wireless Sensor Networks, *24th IEEE Conference on Computer Communications (INFOCOM 2005)*, Miami, FL, pp. 172–183.
- Priyantha, N. B., Balakrishnan, H., Demaine, E. and Teller, S.: 2003, Anchor-Free Distributed Localization in Sensor Networks, *Technical Report TR-892*, MIT Laboratory for Computer Science.
- Reich, J., Misra, V., Rubenstein, D. and Zussman, G.: 2011, Connectivity Maintenance in Mobile Wireless Networks via Constrained Mobility, *30th IEEE Conference on Computer Communications (INFOCOM 2011)*, IEEE, Shanghai, China, pp. 927–935.
- Shell, D. A. and Matari, M. J.: 2004, Directional Audio Beacon Deployment: an Assistive Multi-Robot Application, *IEEE/RSJ International Conference on Robotics and Automation (ICRA 2004)*, New Orleans, LA, pp. 2588–2594.
- Sichitiu, M. L. and Ramadurai, V.: 2004, Localization of Wireless Sensor Networks with a Mobile Beacon, *1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004)*, IEEE, Fort Lauderdale, FL, pp. 174–183.
- Tasaki, T., Tokura, S., Sonoura, T., Ozaki, F. and Matsuhira, N.: 2010, Mobile Robot Self-Localization Based on Tracked Scale and Rotation Invariant Feature Points by Using an Omnidirectional Camera, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, IEEE, Taipei, Taiwan, pp. 5202–5207.