Robust UWB-Based Localization with Application to Automated Guided Vehicles

Stefania Monica and Gianluigi Ferrari*

Herein, localization framework based on ultra wide band (UWB) signaling is described. The proposed framework tackles the issues of industrial indoor localization. In particular, experiments are performed using the PulsON 410 Ranging and Communication Modules (P410 RCMs) produced by Time Domain. The considered scenario consists of an industrial warehouse with laser guided vehicles (LGVs), manual forklifts, and people. In the considered localization setup, three RCMs are placed on an LGV and an additional RCM acts as a Target Node (TN), whose position needs to be estimated. As the laser guarantees perfect knowledge of the LGV position inside the building, the three RCMs on board can be considered as Anchor Nodes (ANs) with known positions. The TN is supposed to be placed on a manual forklift or in the belt of a person inside the warehouse. Two localization algorithms that rely on the distance estimates between pairs of UWB sensors positioned on the automated guided vehicles (AGV) and the UWB sensor that acts as TN are proposed. Extensive performance results based on an experimental campaign performed in an industrial warehouse is also shown, for different TN positions around the LGV.

1. Introduction

In the late 1980s, ultra wide band (UWB) signals started being used in the military field. From then on, the interest in the use of UWB signals increased, especially after 2000 when the Federal Communications Commission (FCC) in the USA allowed the unlicensed use of UWB devices. In 2007, the IEEE 802.15.4a standard was defined, providing physical layer specifications for short-range UWB-based communications.[1]

UWB signaling is an attractive option when dealing with indoor communications.[2] As a matter of fact, the high time resolution and the large frequency spectrum, typical of such signals, improve their capability of penetrating through obstacles and reduce (from a theoretical viewpoint) the impact of interference phenomena such as Non-Line-of-Sight (NLoS) propagation and/or multi path components.[3] Moreover, thanks to their large bandwidth, the duration of the pulses transmitted by UWB systems is very short (on the order of nanoseconds). This guarantees an accurate Time of Flight (ToF) estimation of signals traveling between pairs of nodes.[4] In contrast, in narrow-band systems, pulses received via multiple paths easily overlap leading to wrong ToF estimates and, hence, to inaccurate distance estimates.[1] Another advantage of UWB signaling is related to energy consumption, which is low thanks to the low transmission power of such systems. As a matter of fact, since UWB signals occupy a very large bandwidth, emission power constraints are necessary to avoid interference problems with other devices operating in the same frequency spectrum.[5]

Due to the aforementioned reasons, UWB systems are seen as good candidates to address the issues related to indoor localization and tracking in many fields, such as: assisted living,[6,7] security area surveillance,[8] medical monitoring,[9,10] smart homes,[11,12] localization and tracking of people, vehicles, goods.[13,14] As UWB signals have very large bandwidth, the ToF of signals traveling between pairs of nodes is (in principle) extremely accurate, leading to a good estimate of pairwise distances between nodes. Such distance estimates are the key ingredients to perform accurate localization.

In this article, we focus on the application of the UWB technology to locate targets inside industrial buildings, with the final aim of improving security in scenarios where laser guided vehicles (AGVs) move. LGVs are a particular type of automated guided vehicles (AGVs) that use lasers for navigation. Nowadays, the role of LGVs is becoming more and more important in the design of new factories and warehouses. In automated processes, LGVs are typically used to move goods around warehouses and are programmed to communicate with each other to ensure that products are moved safely through the factory. An LGV carries a laser transmitter/receiver on a rotating turret and its navigation relies on the use of reflective tapes that can be placed, for instance, on the walls and/or on the ceiling of the warehouse.

Dr. S. Monica
Department of Sciences and Methods for Engineering
Via Amendola 2, 42122 Reggio Emilia, Italy
E-mail: stefania.monica@unipr.it

Dr. G. Ferrari
Internet of Things (IoT) Lab
Department of Engineering and Architecture
Parco Area delle Scienze 181/A
43124 Parma and CNIT Research Unit of Parma
Parco Area delle Scienze
181/A, 43124 Parma, Italy

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/aisy.202000083.

DOI: 10.1002/aisy.202000083

The angle and the distance to any reflector in Line of Sight (LoS) and within range are automatically calculated and used to estimate the current position of the AGV using a proper triangulation algorithm. In this way, the position of each LGV is always known and the localization accuracy (due to the laser resolution) is typically of the order of 1 mm.[15] The logistics of all the LGVs that move inside the same warehouse is handled by a supervising system, which also guarantees that LGVs do not clash. In contrast, there is no way of avoiding collisions between LGVs and manual forklifts or people in the same area, as objects and people are not included in the LGV overall supervising system. For this reason, it is of interest to investigate the performance of UWB-based localization to design and implement a low-cost radio communication localization system.

The literature on indoor localization inside industrial buildings includes several works and the use of many different localization systems has been investigated in this context. One of the approaches for indoor localization in industrial scenarios involves the use of cameras.[16,17] Camera-based systems can cover a large indoor area and they can provide a localization accuracy in the order of centimeters or decimeters, depending on how the systems are configured and on the specific application scenario.[18] Even though the localization accuracy obtained with camera-based systems is good enough for many applications, these kind of systems have some drawbacks. As a matter of fact, the light conditions and the possible darkness in some areas of the industrial warehouse can impact on the localization accuracy. Moreover, to use the approach described in the literature on this topic, several cameras need to be placed inside the considered building and, hence, this approach can be expensive in terms of installation cost. At the opposite, the approach described in this article requires to install sensors on the LGVs and on target, without the need to wire the entire building, thus avoiding placing a large number of sensors inside the (possibly large) indoor industrial environment. Finally, cameras need to be in LoS with the considered target to be used, whereas other technologies (such as UWB) can work also if LoS is not guaranteed. Other works on indoor localization focus on the use of sound propagation,[19] but such systems are practically unfeasible in industrial environments, which can be very noisy. The use of the infrared technology has also been investigated, but it is typically used for presence detection rather than for position estimates acquisition.[20] Indoor localization systems based on magnetic field have also been proposed in the literature,[21] but they can have several drawbacks in indoor industrial environments. First, the magnetic field can be perturbated by the presence of electric motors operating inside the same building. Moreover, the localization error obtained when using these systems is typically in the order of a few meters and it can be too inaccurate for the application addressed in this article. Another approach to localization involves the use of inertial sensors.[22] However, when using inertial systems each position estimate is calculated not only on the basis of the data measured by the sensors but also on the basis of the previously obtained position estimates. For this reason, all inertial systems suffer from drift and, hence, they can only be used as complementary sensors in addition to other localization technologies. Another technology that could be used in the context of indoor localization is radio-frequency identification (RFID).[23] The localization accuracy of RFID systems depends on the number and on the dislocation of tags inside the considered indoor environment and a large number of tags is needed to guarantee accurate localization in large environments, such as industrial warehouses, thus increasing the installation cost. Moreover, position estimates acquired via RFID typically have an accuracy of decimeters or meters, which may not be sufficient for some applications. WIFI systems have also been studied for localization purposes.[24] In this case, distance estimates between tags and access points are derived from the Received Signal Strength (RSS) of signals traveling between nodes and, for this reason, they can be very inaccurate, especially in the presence of obstacles inside the environment, thus leading to inaccurate position estimates. For this reason, the accuracy of these systems is typically not sufficient to manage localization in large industrial scenarios. Other RF systems such as Zigbee and Bluetooth, have been studied in the context of localization. The performance and the applicability of such technologies can vary, depending on the chosen technology, on the pervasiveness of devices, and on the installation of the infrastructure inside the building. However, according to various studies on the topic, UWB is the most promising technology that can be used for localization inside large industrial environments and this is the reason why we focus on the use of UWB in the remaining of the article.

In this article, we propose a UWB-based localization system that allows locating Target Nodes (TNs) close to the LGVs. The performance of the proposed system is tested in a scenario with static TN, but the proposed approach is general and could be applied also with a moving TN and/or moving LGV. Experimental results shown in the last part of the article are expedient to evaluate the performance of the proposed localization algorithms, in terms of the errors on the position estimates of the TN all around the LGV. More precisely, our aim is to investigate in which areas around the LGV the TN position estimates are sufficiently reliable and to identify areas in which they are not. In the considered setup, we assume that each LGV is equipped with a small number of UWB modules placed on its top at known positions (with reference to the LGV's on-board reference system). Under this assumption, as the (global) position of each LGV can be considered as known (thanks to the accurate laser positioning system), the modules on each LGV become "Anchor Nodes" (ANs) with known positions. Such ANs can then be used to localize TNs in various positions around the LGV. We remark that the majority of the studies presented in the literature use fixed ANs installed in the warehouse to locate TNs (see, e.g., ref. [25]). In contrast, the approach proposed in this article assumes that ANs are placed on the LGVs, so that each LGV has its own reference system.

All the results presented in this article have been derived on the basis of an experimental campaign that has been carried out inside an industrial warehouse. In the considered scenario, TNs could be people and manual forklifts situated in the same indoor environment—this is a typical situation in industrial warehouses. Our approach assumes that the TN is equipped with a UWB module able to communicate, on the UWB channel, with UWB modules placed on the LGV. We remark that our approach allows avoiding wiring the entire warehouse, as required by other commercial solutions.[26] Each AN on the LGV acquires, through the UWB channel, the range estimates from the TN and communicates these data to a Raspberry Pi, which is connected to all the

ANs on the LGV. Localization is then performed using two distance-based TN localization algorithms running on the Raspberry Pi as will be explained in the following sections. The first (starting) algorithm is based on the well-known Circumference Intersection (CI) algorithm and has been adapted to the specific application scenario, to tackle issues related to errors on distance estimates and/or to possible unavailability of distance estimates from one of the ANs. The second (final) algorithm is an improved version of the first one, as it introduces proper error mitigation strategies. Details on both algorithms are given in Section 4. The accuracy of the proposed localization algorithms is investigated with various TN' positions situated all around the LGV. This is meant to study whether TNs can be identified in all the possible different positions close to the LGV. The performance of the proposed indoor localization system is investigated using the UWB sensors PulsON 410 Ranging and Communications Modules (P410 RCMs) produced by Time Domain.

The article is organized as follows. Section 2 outlines the main features of the UWB sensors that have been used in our experiments. In Section 3, the setup in the considered scenario is described. Section 4 describes the implemented localization approaches. In Section 5, the experimental results are shown. Section 6 concludes the article.

2. The PulsOn410 RCMs

The experimental results shown in this article have been obtained using the P410 RCMs produced by Time Domain. The P410 RCMs are single-board UWB radio components intended to be integrated into users' electronic devices to provide accurate estimates of inter-node distances at a high update rate. The distance estimate provided by the RCMs is denoted as Precision Range Measurement (PRM).[27]

Usually, the communication architecture encompasses many P410 RCMs and at least one host, connected to at least one module via USB connection. Each RCM can communicate, on the UWB channel, to all the RCMs in the considered environment, and each message exchange is associated with a message identifier. Each RCM has an ID and communications between pairs of RCMs can be carried out either as broadcast or by directly enquiring a given ID. After the reception of a UWB packet, each RCM connected to any host communicates all the received informations (through proper libraries provided by Time Domain) to its host, which can process all the data.

A detailed list of the data structures that the C library provides can be found in the official Time Domain documentation.[28] The main C library function that we used to get range informations from a TN is rcmRangeToEx(), which is a modified version of the function rcmRangeTo(), provided with the documentation of the RCM, that also takes the ID of the requester node as an additional input parameter. The reason for this change is motivated by the localization strategy and will be clarified in the following. At the end of a full UWB ranging conversation, the requesting RCM transmits the received ranging data to its host, using the message rcmMsg_RangeInfo. For the sake of simplicity, we simply outline the parameters of the message rcmMsg_RangeInfo that are strictly necessary to perform localization and that will be used in the following:

responderId: ID of the responding module;

precisionRangeMm: distance (dimension: [mm]) between a pair of UWB modules based on a TW-ToF measurement (PRM);

vPeak: the absolute maximum value in the leading edge window of the received waveform, where the leading edge corresponds to the highest energy impulse of the received signal.

We remark that vPeak is a quality metrics that accompanies any range message: it can be described as an improved form of the RSS that rarely suffers from multipath fading.[28] The vPeak is a measure of the maximum absolute value of the amplitude measured just after the leading edge offset.

3. Scenario and Notation

In this section, we describe the setup considered to obtain the experimental results shown in Section 4. The considered indoor environment is a warehouse and the localization framework involves one LGV (on which RCMs, acting as ANs, are placed) and one TN (which consists of a RCM). In **Figure 1**, a picture of the considered LGV is shown. Throughout the article, we refer to the part with the forks as the "back" of the LGV and to the part without the forks as the "front" of the LGV. Observe that the forks in the back are taller than the LGV itself, originating non-negligible NLoS phenomena. The used LGV is equipped with three RCMs used as ANs. In our experimental setup, such three RCMs are placed on top of the LGV and are all connected to the same host, namely a Raspberry Pi, on which the proposed localization algorithm (described in Section 4) is implemented. The Raspberry Pi communicates with the computer on board of the LGV via Ethernet connection, so that the localization data can be transmitted to the LGV and properly processed. In **Figure 2**, the projection of the LGV on the xy-plane is shown (black rectangle), together with the considered positions of the three RCMs AN_0, AN_1, and AN_2 (colored stars) and the Raspberry Pi, denoted as RPi.

Multiple distance measurements are used to estimate the TN's positions, also in the presence of NLoS phenomena due to the (possible) presence of part of the LGV (e.g., forklifts) between one or more ANs and the TN. The ANs and the TN are positioned on the same plane. We remark that the same algorithm could also be applied if the TN was positioned at a different height, provided that the latter was known (or, at least, estimated). As a matter of fact, in such a scenario the range estimates could be corrected according to the Pythagorean theorem, thus obtaining the projections of such range estimates on the plane where the ANs lie. Observe that, under this assumption, three is the minimum number of ANs required to estimate the TN's position using the distance estimates between each AN and the TN. More precisely, three range measurements from the three ANs identify three circumferences, centered in the ANs' positions and with radii corresponding to the (estimated) AN-TN distances. Should the true pairwise distances between nodes be known, the intersection of such circumferences would be a single point—namely, the TN position. In practice, as the range measurements are affected by errors due to wireless

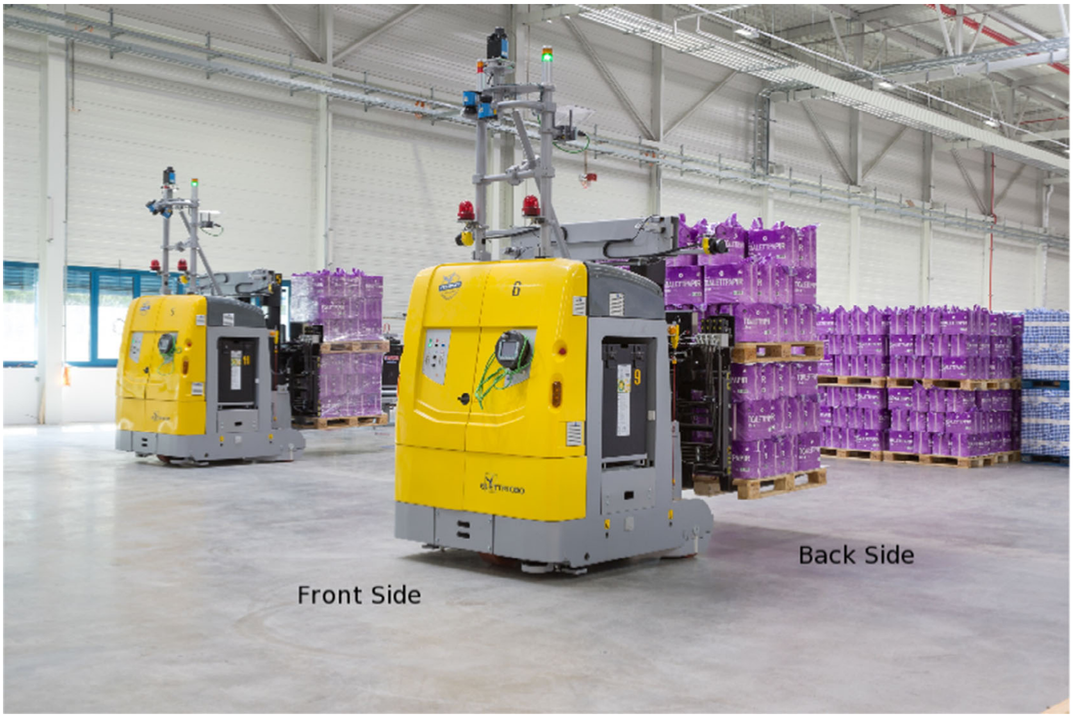

Figure 1. Picture of LGVs in the considered industrial scenario.

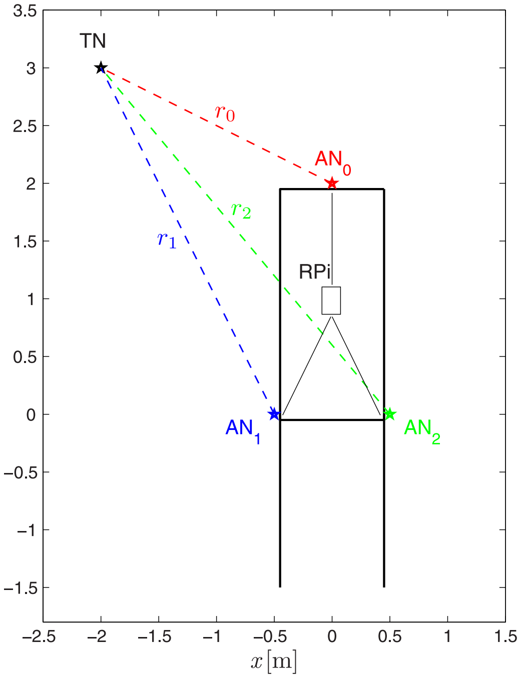

Figure 2. The LGV projection (black rectangle), the ANs (colored stars) positions, and possible TN position (black star) are shown. The ANs–TN distances $\{r_i\}_{i=0}^{2}$ are also shown (dashed line).

propagation, the radii are likely to be erroneous. Therefore, proper localization algorithms are necessary to cope with range estimation errors.

With reference to Figure 2, we denote as AN$_0$ the sensor in the front of the LGV and as AN$_1$ and AN$_2$ the sensors in the back of the LGV. We remark that the choice of having two RCMs at the back of the LGV is motivated by the presence of forks, which increases the impact of NLoS-related interference phenomena. However, the localization algorithms proposed in the following are general and they can be applied with different configurations of ANs on the LGV. In Figure 2 and in the following derivation, we consider a 2D coordinate system centered in the barycenter of the two positions of AN$_1$ and AN$_2$, with AN$_0$ laying on the positive y-axis. We remark that, in our experiments, the true height of the RCMs on the LGV corresponds to the height of the LGV, namely 2 m. To simplify the notation, as all the RCMs lie on a plane (namely, $\{z = 2\mathrm{m}\}$), their coordinates are denoted as vectors of length 2, thus considering only the x and y components. In particular, the coordinates of the ANs in the considered coordinate system are denoted as

$$\underline{s}_0 = [0, L] \quad \underline{s}_1 = [-V, 0] \quad \underline{s}_2 = [V, 0] \tag{1}$$

where L and V are system parameters associated with the length and the width of the LGV. In our experiments, the values of such parameters are: $L = 2$ m and $V = 0.5$ m. Figure 2 also shows a representative TN position and its distances $\{r_i\}_{i=0}^{2}$ (dashed lines) from $\{\mathrm{AN}_i\}_{i=0}^{2}$. Denoting as $\underline{u} = [x, y]$ the true position of the TN (on the plane $\{z = 2\mathrm{m}\}$), the distances $\{r_i\}_{i=0}^{2}$ can be expressed as $r_i = ||\underline{s}_i - \underline{u}||$.

Multiple distance measurements are used to estimate the TN's positions, also in the presence of NLoS phenomena due to the (possible) presence of part of the LGV (e.g., forklifts) between one or more ANs and the TN.

In the following, we consider various TN positions around the LGV. For each of the TN configurations, N_s (independent) range estimates from each AN are taken, thus leading to N_s position estimates of the considered TN, which are denoted as $\underline{u}^{(j)} = [\hat{x}^{(j)}, \hat{y}^{(j)}]$, $j \in \{1, \ldots, N_s\}$. Then, the distance (namely, the error) between the true TN position and its estimate in the $j-$th iteration can be expressed as

$$d^{(j)} \triangleq \sqrt{(x - \hat{x}^{(j)})^2 + (y - \hat{y}^{(j)})^2} \tag{2}$$

By averaging over the N_s iterations, the following average distance is defined

$$d_{\text{avg}} \triangleq \frac{1}{N_s} \sum_{j=1}^{N_s} d^{(j)} \tag{3}$$

4. Localization Algorithms

Considering the scenario in Figure 2, the TN position $\underline{u}$ could be found as the intersection of the circumferences $\{\mathcal{C}_i\}_{i=0}^{2}$ centered in $\{\underline{s}_i\}_{i=0}^{2}$ with radii $\{r_i\}_{i=0}^{2}$ equal to the true distances between the TN and the $i-$th AN, i.e., by solving the following system

$$\begin{cases} \mathcal{C}_0 : x^2 + (y - L)^2 = r_0^2 \\ \mathcal{C}_1 : (x + V)^2 + y^2 = r_1^2 \\ \mathcal{C}_2 : (x - V)^2 + y^2 = r_2^2 \end{cases} \tag{4}$$

As the true distances $\{r_i\}_{i=0}^{2}$ are unknown, localization must be based on their estimates. Let us denote as $\{\hat{r}_i^{(j)}\}_{i=0}^{2}$ the estimates of $\{r_i\}_{i=0}^{2}$ in the $j-$th iteration, $j \in \{1, \ldots, N_s\}$. Then, the position estimate $\underline{u}^{(j)}$ in the $j-$th iteration has to be determined by relying on the circumferences, which are denoted as $\{\mathcal{C}_i^{(j)}\}_{i=0}^{2}$, obtained by substituting the true distances $r_i^{(j)}$ with the estimated distances $\hat{r}_i^{(j)}$ in Equation (4).

Two localization algorithms are described in the remainder of this section: the first one is meant to be the basis on which the second (final) one builds upon. The first one only relies on the estimated distances between the ANs and the TN and is an improved version of the CI algorithm described in ref. [29]. The second one can be considered as an improvement of the first one, as it uses the values of vPeak, provided by the RCMs, to decide if a distance measurement is reliable or not. Both algorithms can be divided into two steps: the first one corresponds to the acquisition of the distance estimates between each AN and the TN; in the second one, localization is performed on the basis of the data collected in the first step.

The algorithms are explained in detail in the two following subsections. For the sake of simplicity, while explaining the algorithms we omit the superscript (j), used to denote the $j-$th iteration, and we simplify the notation writing $[\star]$ instead of $[\hat{\star}]$ even if the considered quantities are estimates. We remark that this notation does not introduce ambiguity because the algorithms makes use of estimated quantities.

4.1. A Distance-Based Localization Algorithm

In this subsection, a distance-based localization algorithm is explained. First, the distance estimates between each AN and the TN are acquired. The nodes AN_1 and AN_2 are put in sleep mode and the node AN_0 performs a range request in broadcast, so that each RCM within range (but AN_1 and AN_2) can reply. As soon as a node (namely, a TN) replies, the obtained distance estimate precisionRangeMm is saved in r_0 and the ID of the replying node responderId is passed to AN_1 and AN_2, which are woken up one at a time to directly interrogate the acquired TN. Observe that, due to the LGV itself, AN_1 and/or AN_2 may be in NLoS with the RCM that replied to AN_0. Due to propagation errors, it is then possible that the enquired TN does not reply to the range request from AN_1 and/or AN_2. Two specific variables, denoted as status_1 and status_2, are defined to deal with these cases. More precisely, status_1 and/or status_2 are kept equal to 0 if the range estimates from AN_1 and/or AN_2 are not available, otherwise they are set to 1 and the distance estimates are saved in r_1 and/or r_2, respectively.

After the range estimate acquisition phase, localization is performed on the basis of the previously obtained estimates. Depending on the number of available range estimates, different localization strategies are considered. The localization strategy often requires to identify the two nearest points chosen from two sets composed of two points. To this aim, we define the function findNearer, which takes as inputs two sets (say A and B), each of which contains the coordinates of two points, and returns the coordinates associated with the two nearest points from the two sets, namely $\underline{p} \in A$ and $\underline{q} \in B$ that satisfy

$$||\underline{p} - \underline{q}|| = \min_{\underline{a} \in A,\, \underline{b} \in B} ||\underline{a} - \underline{b}|| \tag{5}$$

Moreover, the proposed algorithms often require to deal with pairs of points and to identify which of the two points has the minimum or the maximum abscissa. To this aim, we define the functions maxAbscissa and minAbscissa, which take as input a set containing the coordinates of two points and return the coordinates of the point with the largest/smallest abscissa, respectively.

We can now describe the localization phase of the considered algorithm. We remark that every time the TN position $\underline{u}$ cannot be estimated, we set $\underline{u} = [\infty, \infty]$. Four cases are identified, on the basis of the number of available distance estimates.

```
procedure findTargetPosition

$I \leftarrow \{\underline{p}_0, \underline{p}_1\} = \mathcal{C}_0 \cap \mathcal{C}_1$
$J \leftarrow \{\underline{q}_0, \underline{q}_1\} = \mathcal{C}_0 \cap \mathcal{C}_2$
if($I \neq \varnothing \wedge J \neq \varnothing$)
    $\{\underline{p}, \underline{q}\} \leftarrow$ findNearer($I, J$), $\underline{u} \leftarrow \frac{1}{2}(\underline{p} + \underline{q})$
else if($I = \varnothing \wedge J \neq \varnothing$)
    $A \leftarrow \mathcal{C}_0 \cap \mathcal{R}_{01}$, $B \leftarrow \mathcal{C}_1 \cap \mathcal{R}_{01}$
    $I = \{\underline{p}_0, \underline{p}_1\} \leftarrow$ findNearer($A, B$)
    if ($||\underline{p}_0 - \underline{p}_1|| \le d_{Th}$)
        $\{\underline{p}, \underline{q}\} \leftarrow$ findNearer($I, J$), $\underline{u} \leftarrow \frac{1}{2}(\underline{p} + \underline{q})$
    else
        $\underline{u} \leftarrow$ maxAbscissa($J$)
else if($I \neq \varnothing \wedge J = \varnothing$)
    $A \leftarrow \mathcal{C}_0 \cap \mathcal{R}_{02}$, $B \leftarrow \mathcal{C}_2 \cap \mathcal{R}_{02}$
    $J = \{\underline{q}_0, \underline{q}_1\} \leftarrow$ findNearer($A, B$)
    if ($||\underline{q}_0 - \underline{q}_1|| \le d_{Th}$)
```

```
18)       $\{\underline{p}, \underline{q}\} \leftarrow \text{findNearer}(I, J),\ \underline{u} \leftarrow \frac{1}{2}(\underline{p} + \underline{q})$
19)    else
20)       $\underline{u} \leftarrow \text{minAbscissa}(I)$
21) else if$(I = \varnothing \wedge J = \varnothing)$
22)    $A \leftarrow \mathcal{C}_0 \cap \mathcal{R}_{01},\ B \leftarrow \mathcal{C}_1 \cap \mathcal{R}_{01}$
23)    $I = \{\underline{p}_0, \underline{p}_1\} \leftarrow \text{findNearer}(A, B)$
24)    $A \leftarrow \mathcal{C}_0 \cap \mathcal{R}_{02},\ B \leftarrow \mathcal{C}_2 \cap \mathcal{R}_{02}$
25)    $J = \{\underline{q}_0, \underline{q}_1\} \leftarrow \text{findNearer}(A, B)$
26)    $d_1 \leftarrow ||\underline{p}_0 - \underline{p}_1||,\ d_2 \leftarrow ||\underline{q}_0 - \underline{q}_1||$
27)    if $(d_1 \leq d_{\mathrm{T}h} \wedge d_2 \leq d_{\mathrm{T}h})$
28)       $\{\underline{p}, \underline{q}\} \leftarrow \text{findNearer}(I, J),\ \underline{u} \leftarrow \frac{1}{2}(\underline{p} + \underline{q})$
29)    else if $(d_1 \leq d_{\mathrm{T}h} \wedge d_2 > d_{\mathrm{T}h})$
30)       $\underline{u} \leftarrow \frac{1}{2}(\underline{p}_0 + \underline{p}_1)$
31)    else if $(d_1 > d_{\mathrm{T}h} \wedge d_2 \leq d_{\mathrm{T}h})$
32)       $\underline{u} \leftarrow \frac{1}{2}(\underline{q}_0 + \underline{q}_1)$
33)    else
34)       $\underline{u} \leftarrow [\infty, \infty]$
```

Listing 1: Pseudocode of procedure findTargetPosition.

4.1.1. Case 1

If distance estimates from AN_1 and AN_2 are both available (i.e., $\mathrm{status}_1 = 1$ and $\mathrm{status}_2 = 1$), the localization algorithm is implemented with the procedure findTargetPosition in Listing 1, which can be described as follows. In this case, the three distance measurements $\{r_i\}_{i=0}^{2}$ are available and the final position estimate is obtained starting from the intersections of pairs of circumferences. More precisely, let us define the following two sets (lines 3 and 4 of Listing 1)

$$I = \mathcal{C}_0 \cap \mathcal{C}_1 \quad J = \mathcal{C}_0 \cap \mathcal{C}_2 \tag{6}$$

Each set can either: contain the coordinates of two points if the considered circumferences intersect; be empty, if the circumferences do not intersect.

i) If both pairs of circumferences intersect (line 5 of Listing 1), i.e., I and J are not empty sets, then we choose the two nearest points from I and J, according to the function findNearer, and the final position estimate is selected as the barycenter of these two points (line 6 of Listing 1).

ii) If $\mathcal{C}_0$ and $\mathcal{C}_2$ intersect but I is empty (line 7 of Listing 1), we look for the two nearest points of $\mathcal{C}_0$ and $\mathcal{C}_1$. The reason for doing so is that $\mathcal{C}_0$ and $\mathcal{C}_1$ may not intersect just because of small errors in the estimates of r_0 and r_1. To find the two nearest points of $\mathcal{C}_0$ and $\mathcal{C}_1$, we need to intersect both these circumferences with the line $\mathcal{R}_{01}$ that passes through their centers, whose equation is

$$\mathcal{R}_{01}: y = \frac{L}{V}x + L \tag{7}$$

Each of the two intersections $\mathcal{C}_0 \cap \mathcal{R}_{01}$ and $\mathcal{C}_1 \cap \mathcal{R}_{01}$ returns two points and the two nearest points of $\mathcal{C}_0$ and $\mathcal{C}_1$ can be found according to the function findNearer. If the distance between these points is below a given distance threshold d_{Th} (line 10 of Listing 1), then I is redefined as the set containing these two points and the localization is performed as in the first case. Otherwise, it is reasonable to assume that the distance estimate r_1 is far inaccurate and this may be likely due to the fact that the TN and AN_1 are in NLoS. For this reason, to be coherent with the fact that AN_1 is obscured by the LGV, the final position estimate $\underline{u}$ is defined as the point in J with the greatest abscissa (line 13 of Listing 1).

iii) Similar considerations hold if $\mathcal{C}_0$ and $\mathcal{C}_1$ intersect but J is empty (line 14 of Listing 1). In this case, the two nearest points of $\mathcal{C}_0$ and $\mathcal{C}_2$ can be found by intersecting the two circumferences with the line $\mathcal{R}_{02}$ that passes through their centers, i.e.

$$\mathcal{R}_{02}: y = -\frac{L}{V}x + L \tag{8}$$

If the distance between these two points is below the distance threshold d_{Th} (line 17 of Listing 1), then J is redefined as the set containing the two points and the localization is performed as in the first case. To clarify the aforementioned explanation, in **Figure 3**a, we show an illustrative example relative to this case. Otherwise, as the points in I are symmetric with respect to the line $\mathcal{R}_{01}$ defined in Equation (7), the final position estimate $\underline{u}$ is the point in I with the smallest abscissa (line 20 of Listing 1), to make sure that AN_2 is obscured by the LGV. To make this idea more intuitive, in Figure 3b, an illustrative example relative to this specific case is shown.

iv) Finally, if both I and J are empty (line 21 of Listing 1), we look for the two nearest points of $\mathcal{C}_0$ and $\mathcal{C}_1$ and the two nearest points of $\mathcal{C}_0$ and $\mathcal{C}_2$. If the distances between the two pairs of points are below d_{Th} (line 27 of Listing 1), then I and J are redefined as the sets containing these two pairs of points and the localization is performed as in the first case. If the distance between the two nearest points of $\mathcal{C}_0$ and $\mathcal{C}_1$ is below $d_{\mathrm{T}h}$ and the distance between the two nearest points of $\mathcal{C}_0$ and $\mathcal{C}_2$ is above $d_{\mathrm{T}h}$ (line 29 of Listing 1), the TN's position is estimated as the barycenter of the first pair of points. Similarly, if the distance between the two nearest points of $\mathcal{C}_0$ and $\mathcal{C}_1$ is above $d_{\mathrm{T}h}$ while the distance between the two nearest points of $\mathcal{C}_0$ and $\mathcal{C}_2$ is below $d_{\mathrm{T}h}$ (line 31 of Listing 1), the TN's position is estimated as the barycenter of the second pair of points. Otherwise, the TN position cannot be estimated (line 33 of Listing 1).

4.1.2. Case 2

If the distance estimate from AN_1 is available and the distance estimate from AN_2 is not available (i.e., $\mathrm{status}_1 = 1$ and $\mathrm{status}_2 = 0$), the localization algorithm is described in a procedure named findTargetPositionNoS2. In this case, localization is handled using only range estimates from AN_0 and AN_1. More precisely, the range estimate from AN_2 is not available, we can only consider $\mathcal{C}_0$ and $\mathcal{C}_1$ and their intersection I defined in Equation (6).

If I is not empty, the TN position estimate is the point in I with the smallest abscissa, for the same considerations about line 20 of Listing 1.

If I is empty, we look for the two nearest points of $\mathcal{C}_0$ and $\mathcal{C}_1$ (as done in line 7 of Listing 1). If their distance is below the distance threshold d_{Th} the TN's position is estimated as the barycenter of such points. This choice is due to the fact that the distance estimate from AN_2 is not available and, therefore,

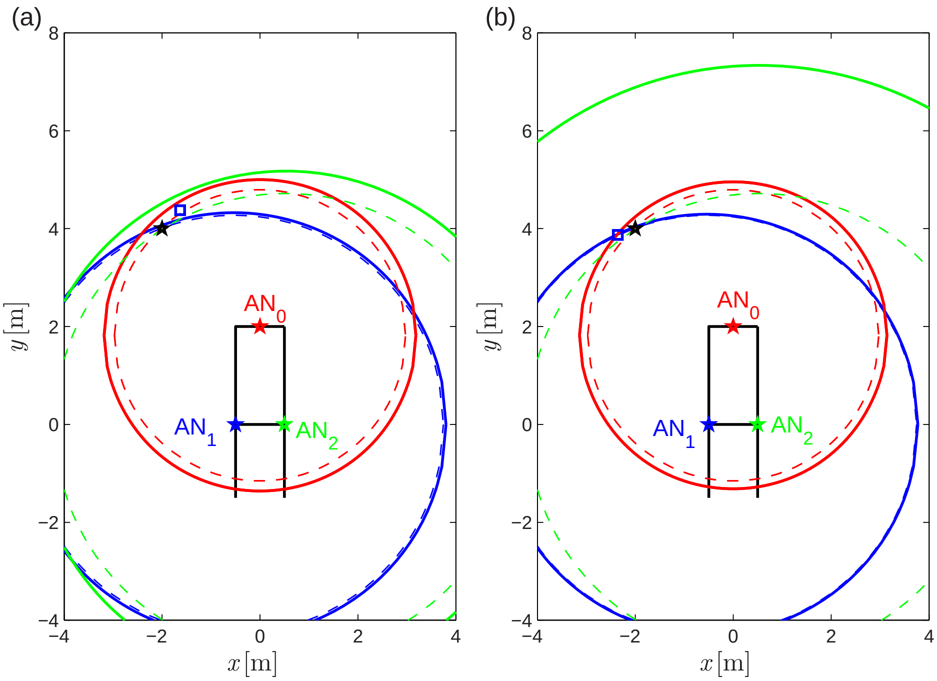

Figure 3. The circumferences $\{\mathcal{C}_i\}_{i=0}^{2}$ (dashed lines) and their estimates: a) line 17 of Listing 1; b) line 20 of Listing 1.

we cannot rely on the set J (unlike in line 7 of Listing 1). Otherwise, the available range estimates do not allow estimating the TN position.

4.1.3. Case 3

If the distance estimate from AN$_1$ is not available and the distance estimate from AN$_2$ is available (i.e., status$_1 = 0$ and status$_2 = 1$), the localization algorithm is described in a procedure named findTargetPositionNoS1. In this case, localization is handled using only range estimates from AN$_0$ and AN$_2$. In this case, we can only consider the intersection J of $\mathcal{C}_0$ and $\mathcal{C}_2$ defined in Equation (6) as the range estimate from AN$_1$ is not available.

If J is not empty, the TN's position estimate is the point in J on the other side of AN$_1$ with respect to the LGV, namely the one with the largest abscissa.

If J is empty and the distance between the two nearest points of $\mathcal{C}_0$ and $\mathcal{C}_2$ is below the distance threshold d_{Th}, then the TN position estimate is defined as the barycenter of the two points. Otherwise, the TN position cannot be estimated.

4.1.4. Case 4

If distance estimates from AN$_1$ and from AN$_2$ are both unavailable (i.e., status$_1 = 0$ and also status$_2 = 0$), the range estimates from AN$_1$ and AN$_2$ are both unavailable. Hence, as only one distance estimate (namely, r_0) is known, a TN's position estimate cannot be obtained. As a matter of fact, knowledge of only one distance estimate does not allow implementing any localization strategy. The only available information is that the TN may lie on a circumference centered in $\underline{s}_0$ with radius r_0.

4.2. A Localization Algorithm Based on Distances and VPeak

In this subsection, an improved version of the distance-based algorithm introduced in Section 4.1 is presented. This algorithm can be considered as an improved version of the previous one, as it relies not only on the range estimates $\{r_i\}_{i=0}^{2}$ but also on the corresponding values of vPeak obtained by the ANs. As explained in Section 3, vPeak measures the RSS and, hence, large values of vPeak are likely to be associated to LoS paths while small values of vPeak are associated with NLoS paths. In the following, we will denote as $\{\text{vPeak}_i\}_{i=0}^{2}$ the values of vPeak relative to $\{\text{AN}_i\}_{i=0}^{2}$. The highest the value of vPeak$_i$, the most reliable is the corresponding distance estimate r_i.

This version of the localization algorithm differs from the algorithm introduced in Section 4.1 because in the localization phase the procedure findTargetPosition is replaced by findTargetPositionVPeak explained in Listing 2. The main difference between the two procedures is that in the latter we ignore range measurements associated with small values of vPeak. More precisely, if both vPeak$_1$ and vPeak$_2$ are above a given threshold vPeak$_{\text{Th}}$, then the localization is performed as in Listing 1. If either vPeak$_1$ or vPeak$_2$ are below the threshold vPeak$_{\text{Th}}$, then the localization is performed relying only on the range estimate from the node with the greatest value of vPeak, regardless of the fact that this is above or below the threshold vPeak$_{\text{Th}}$. If vPeak$_1$ and vPeak$_2$ are both below the threshold vPeak$_{\text{Th}}$, we consider the range estimates from the RCM with the largest value of vPeak (otherwise, the TN's position could not be estimated).

```
procedure findTargetPositionVPeak

if (vPeak$_1$ ≥ vPeak$_{Th}$ ∧ vPeak$_2$ ≥ vPeak$_{Th}$)
    findTargetPosition
else
    if (vPeak$_1$ ≥ vPeak$_2$)
        findTargetPositionNoS2
    else
        findTargetPositionNoS1
```

Listing 2: Pseudocode of procedure findTargetPositionVPeak.

5. Experimental Results

In this section, we show experimental results relative to the scenarios described in Section 3. We remark that even though the performance of the proposed localization algorithm is shown in the presence of a static TN, the proposed algorithm can also be used in the presence of a mobile TN. In particular, as each position is computed on the basis of the UWB signals received at the ANs, mobility of the TN can be supported, provided that the displacement between two consecutive sets of transmissions between the TN and the ANs is limited. This is the case in the considered scenario, where the position estimates are evaluated, approximately, every 80 ms, and the maximum speed of the LGVs is 2 m s^{-1}. This means that the LGV can move at most 0.16 m between consecutive estimates. Hence, by tolerating an error of 0.16 m (which is typically lower than the estimated position error), the proposed algorithms are applicable also in the presence of mobility. A deeper investigation of the case with mobile TNs goes beyond the scope of the current article, but represents an interesting research direction toward tracking (not just localization).

As in Figure 2, the values of the parameters L and V in Equation (1) are set to 2 m and 0.5 m, respectively, so that the coordinates of the ANs, expressed in meters, are

$$\underline{s}_0 = [0, 2] \quad \underline{s}_1 = [-0.5, 0] \quad \underline{s}_2 = [0.5, 0] \tag{9}$$

All the results are obtained with the distance threshold, denoted as d_{Th}, set to 0.5 m and the vPeak threshold, denoted as vPeak$_{\mathrm{Th}}$, set to 5000. The choice of these parameters is based on experimental calibration. We assume that range measurements from the three ANs are used to obtain the position estimates of a static TN. We consider 66 different TN positions, which are shown in **Figure 4**. We only consider TN's positions on the left of the LGV as the performance of the proposed localization algorithms on the right are expected to be the same, for symmetry reasons.

For each of the TN positions in Figure 4, the distance-based algorithm described in Listing 1 is applied with $N_c = 100$ iterations, obtaining 100 position estimates per TN position. Once the 100 position estimates are collected, the knowledge of the true TN position allows evaluating the average distance error d_{avg} defined in Equation (3). Then, the improved version of the localization algorithm described in Listing 1 is applied to the same

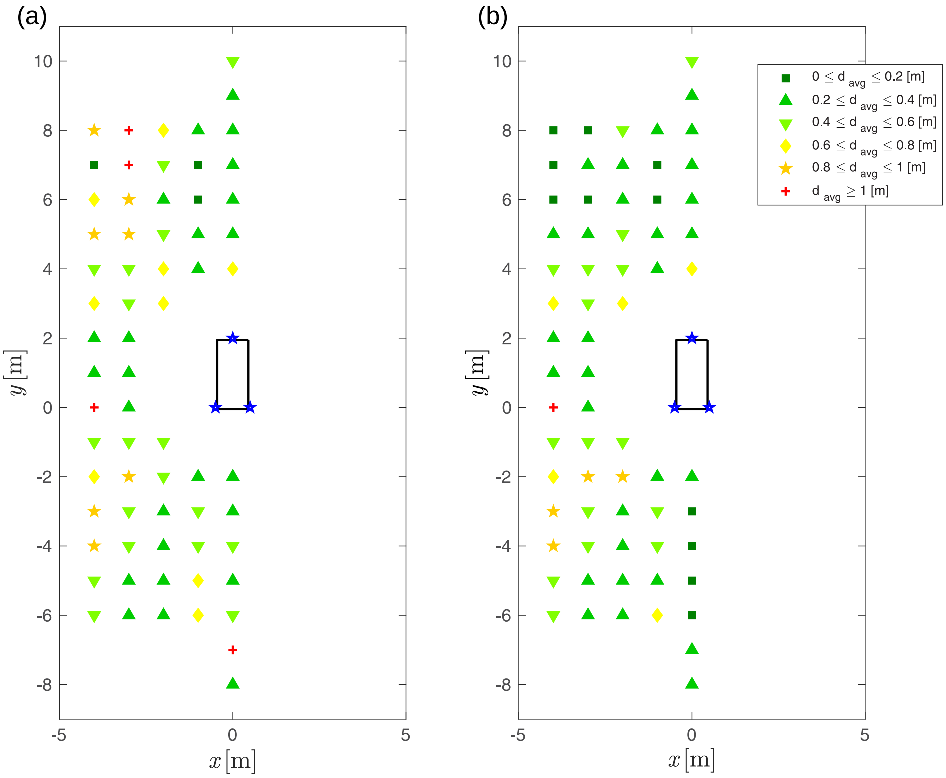

Figure 4. Values of the average distance error d_{avg} relative to different TN positions for a) the distance-based algorithm and b) the hybrid distance/vPeak-based (improved) algorithm.

range estimates used with the former algorithm and the average distance error d_{avg} is evaluated for each TN position.

The obtained experimental results, in terms of d_{avg}, are shown in Figure 4, considering a) the distance-based algorithm and b) the improved algorithm. Each position is associated with a different color, depending on the corresponding value of the average distance error d_{avg}. To be more precise: dark green squares correspond to values of d_{avg} smaller than 20 cm; green triangles correspond to values of d_{avg} between 20 and 40 cm; light green triangles correspond to values of d_{avg} between 40 and 60 cm; yellow diamonds correspond to values of d_{avg} between 60 and 80 cm; orange stars correspond to values of d_{avg} between 80 cm and 1 m; and red pluses correspond to values of d_{avg} larger than 1 m.

From the results in Figure 4 one can conclude that the hybrid distance/vPeak-based (improved) algorithm outperforms the first one. As a matter of fact, Figure 4 a shows that, when using the distance-based algorithm, the value of d_{avg} is below 20 cm only in 3 (out of 66) TN positions, corresponding to 4.5% of the cases. When the improved algorithm is used, instead, the value of d_{avg} is below 20 cm in 11 TN positions, corresponding to 16% of the cases, namely four times the percentage obtained with the first algorithm. Furthermore, d_{avg} obtained with the first algorithm is below 40 cm in 28 TN positions: this corresponds to a percentage equal to 42%, which is nearly ten times the percentage relative to the positions with corresponding d_{avg} smaller than 20 cm. However, in more than half of the considered TNs' positions, the average distance error d_{avg} is greater than 40 cm. On the contrary, when using the improved algorithm, the value of d_{avg} is below 40 cm in 40 TN positions, namely in 60% of the cases.

Moreover, when using the distance-based algorithm, the value of d_{avg} is larger than 1 m in 4 TN positions, while this happens only once with the improved algorithm.

We now discuss, in a comparative way, on the two algorithms. Even if the improved algorithm performs better than the first one, a comparison between Figure 4a,b shows that there is one TN position (namely, $\underline{u} = [-2, -2]$) where the value of d_{avg} obtained with the second algorithm is greater than that obtained with the first one. By analyzing the collected data, we can state that this is due to the fact that, in this case, some range estimates are accurate even if they are associated with small values of vPeak. However, in many (namely, 49) TN's positions, the value of d_{avg} obtained with the distance-based algorithm is similar to that obtained with the improved algorithm (for instance, in the TN positions on the left of the LGV). In 16 positions, the value of d_{avg} obtained with the improved algorithm is smaller than that obtained with the first algorithm: in some of these positions, the improved performance of the second algorithm is particularly evident.

As an illustrative example, let us consider the TN's position with coordinates, expressed in meters, $\underline{u} = [-3, 8]$. In this case, the improvement of the second algorithm is particularly noticeable as it allows reducing d_{avg} below 20 cm (dark green square in Figure 4b) while, according to the first algorithm, d_{avg} is larger than 1 m (red plus in Figure 4a). In **Figure 5**, all the 100 position estimates (over consecutive iterations) of the TN placed in $\underline{u} = [-3, 8]$ are shown considering a) the distance-based algorithm and b) the improved algorithm. From Figure 5a, it can be observed that the position estimates with the first algorithm can be very inaccurate. To be precise, 16 position estimates (those clustered to the right) are nearly 9 m from the true TN position (shown as a black star). This is due to the fact that, in this scenario, AN_2 is in NLoS with the TN, leading to wrong estimates of the distance between AN_2 and the TN which have a detrimental impact on the final position estimate.

Figure 6, shows, through a specific example, the reason for the potential high inaccuracy of the localization algorithm based only on distance estimates. Figure 6 shows the circumferences $\{\mathcal{C}_i\}_{i=0}^{2}$ centered in $\{\underline{s}_i\}_{i=0}^{2}$ with radii $\{r_i\}_{i=0}^{2}$ are shown (dashed lines) together with the circumferences $\{\hat{\mathcal{C}}_i^{(29)}\}_{i=0}^{2}$ obtained with the

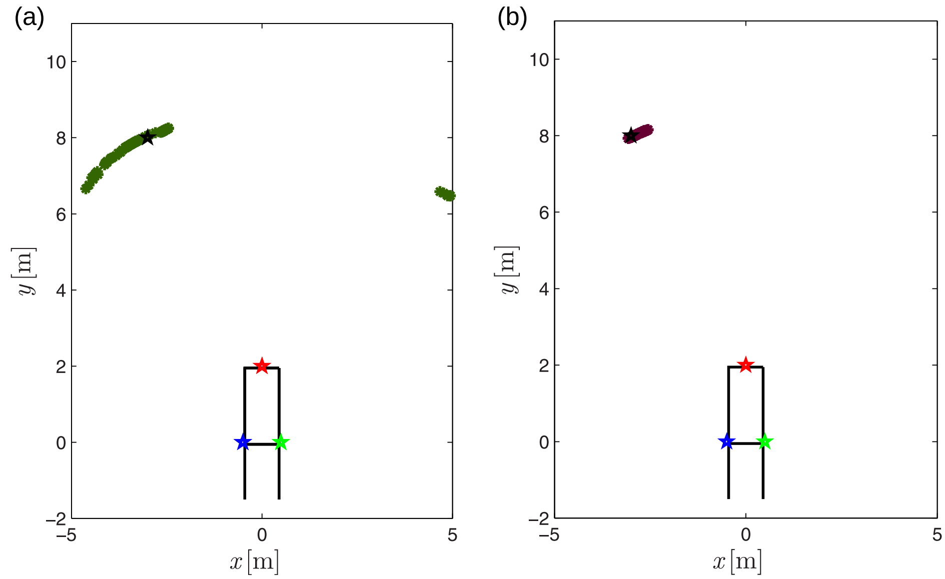

Figure 5. TN position estimates with a) the distance-based algorithm and b) the hybrid distance/vPeak-based (improved) algorithm. The true TN position is shown as a black star.

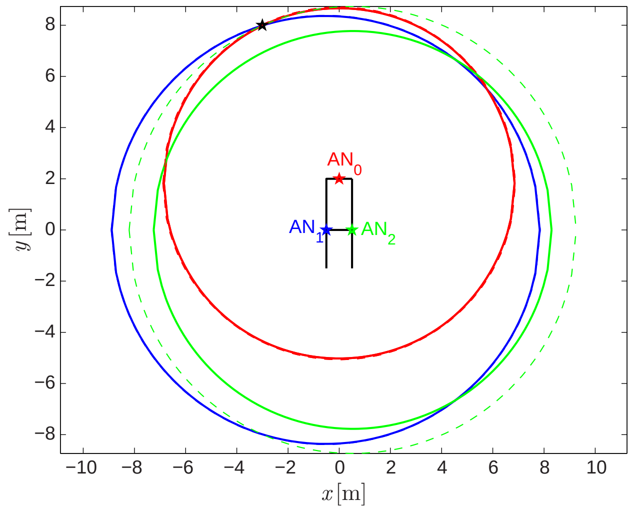

Figure 6. The circumferences $\{\mathcal{C}_i\}_{i=0}^{2}$ (dashed lines) and their estimates obtained from the range estimates in the 29th iteration $\{\hat{\mathcal{C}}_i^{(29)}\}_{i=0}^{2}$ (solid lines) are shown. The coordinates of the TN position estimate in this case, expressed in meters, are.[5.73,5.63]

range estimates $\{\hat{r}_i^{(29)}\}_{i=0}^{2}$ in the 29th iteration. It can be observed that the estimates of r_0 and r_1 are sufficiently accurate and the TN is near one of the two points in $\hat{\mathcal{C}}_0^{(29)} \cap \hat{\mathcal{C}}_1^{(29)}$. On the contrary, r_2 is underestimated by nearly 1 m. Due to this fact, the two points in correspondence to which $\hat{\mathcal{C}}_2^{(29)}$ intersects $\hat{\mathcal{C}}_0^{(29)}$ are far from the true TN position and, moreover, one of them is near the wrong intersection between $\hat{\mathcal{C}}_0^{(29)}$ and $\hat{\mathcal{C}}_1^{(29)}$. The procedure findTargetPositionVPeak used in the improved algorithm can overcome this problem. As a matter of fact, in the 29th iteration the value of vPeak$_2$ is below the threshold vPeak$_{\text{Th}}$, so that the localization estimate is performed ignoring the range estimate from AN$_2$ and choosing, as a final position estimate, the point in $\hat{\mathcal{C}}_0^{(29)} \cap \hat{\mathcal{C}}_1^{(29)}$ with the smallest abscissa, namely the one which is actually nearer to the TN.

In **Figure 7**, the performance of the proposed localization strategies is analyzed in terms of the values of the distances $\{d^{(j)}\}_{j=1}^{100}$ between the true TN position and its estimates, in correspondence to two illustrative TN's positions. More precisely, in Figure 7, the values of the distances $\{d^{(j)}\}_{j=1}^{100}$ are shown as a function of the number of iterations, obtained when applying the

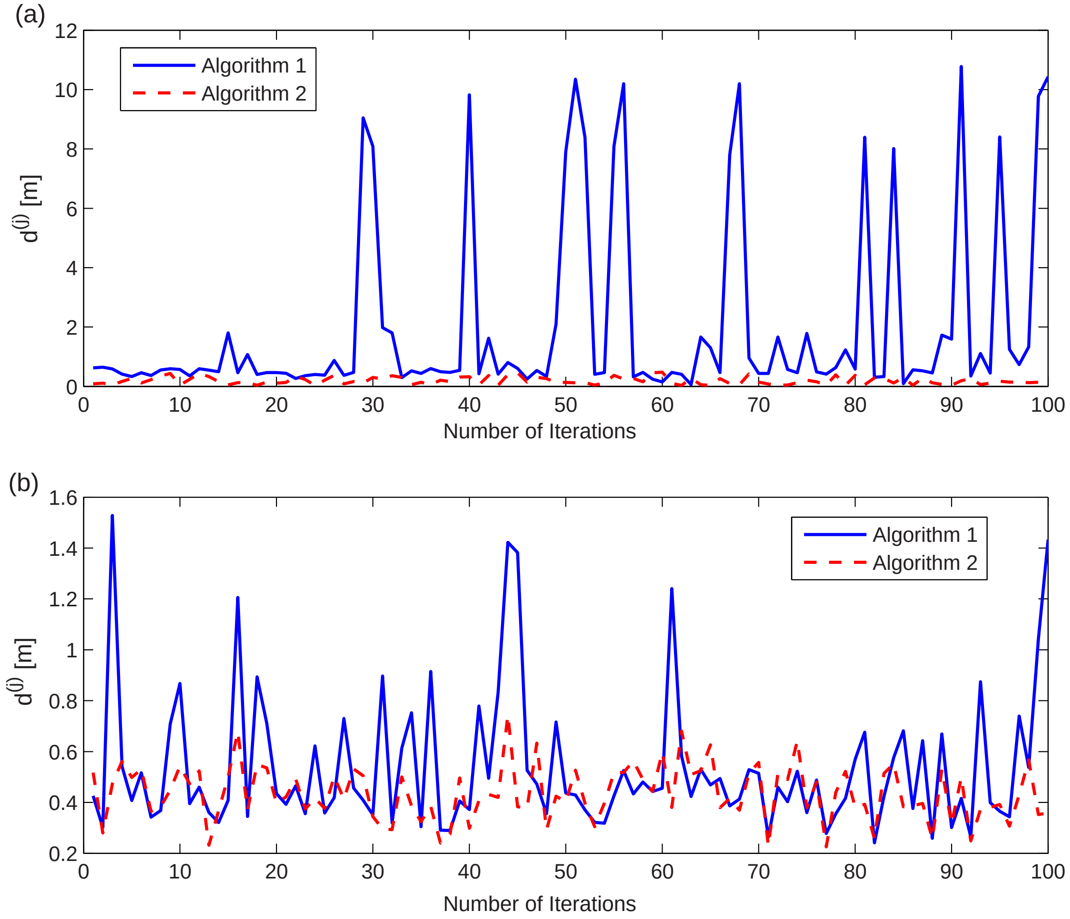

Figure 7. Values of $d^{(j)}$, as a function of the number of iteration $j \in \{1, \ldots, 100\}$, in correspondence to two TN positions: a) $\underline{u} = [-3, 8]$, b) $\underline{u} = [-1, -3]$. In both cases, the first algorithm (solid blue lines) and the improved algorithm (dashed red lines) are considered.

proposed localization algorithms: a) $\underline{u} = [-3, 8]$; and b) $\underline{u} = [-1, -3]$. Solid blue lines refer to the values of $\{d^{(j)}\}_{j=1}^{100}$ obtained with the algorithm based only on distance estimates, whereas dashed red lines refer to the values of $\{d^{(j)}\}_{j=1}^{100}$ obtained with the improved algorithm. Concerning Figure 7a, it can be observed that the position estimates obtained with the first algorithm can be far inaccurate, leading to large values of $d^{(j)}$ for some values of j. More precisely, the maximum value of $d^{(j)}$ is equal to 10.77 m and the value of d_{avg} (according to Equation (3)) is 2.02 m. On the contrary, the position estimates obtained with the second algorithm are sufficiently accurate for the considered application, as they lead to an average distance d_{avg} equal to 0.18 m, with a maximum value of $d^{(j)}$ equal to 0.47 m. The performance improvement of the second algorithm relies the fact that considering the values of vPeak allows discarding inaccurate range estimates from AN_2 that would lead to inaccurate position estimates of the TN. In addition, the range estimates from AN_2 are also associated with low values of vPeak and, hence, the threshold vPeak_{Th} used in the improved algorithm allows discarding such range estimates, thus obtaining more accurate position estimates. As a matter of fact, in the considered scenario the values $\{\hat{r}_2^{(j)}\}_{j=1}^{100}$ may be affected by errors due to NLoS between AN_2 and the TN positioned in $\underline{u} = [-3, 8]$. Figure 7b shows the values of $\{d^{(j)}\}_{j=1}^{100}$ relative to the case with the TN positioned in $\underline{u} = [-1, -3]$. Also in this case, the mitigation strategy applied in the second algorithm allows improving the performance of the first algorithm based only on distance estimates, even though the difference between the performance with the first algorithm (solid blue line) and the second algorithm (dashed red line) is less remarkable than the one in Figure 7a. The maximum values of the distances $\{d^{(j)}\}_{j=1}^{100}$ when applying the first algorithm and the improved one correspond to 1.52 and 0.47 m, respectively, leading to values d_{avg} equal to 0.53 and 0.43 m, respectively. The application of the improved algorithm allows discarding the range estimates from AN_0 (namely, the sensor that is not in LoS with the TN) associated with low values of vPeak. Because of NLoS phenomena, such range estimates are inaccurate, and, hence, degrade the performance of the distance-based algorithm.

Figure 8 shows the distribution of the distance error $\{d^{(j)}\}_{j=1}^{100}$ corresponding to the TN positioned in $\underline{u} = [-3, 7]$ when applying the distance-based algorithm (upper figure) and the improved one. More precisely, Figure 8 shows the number of position estimates corresponding to different values of the distance error $\{d^{(j)}\}_{j=1}^{100}$. It can be observed that when considering the distance-based algorithm (upper figure), the majority of the values of

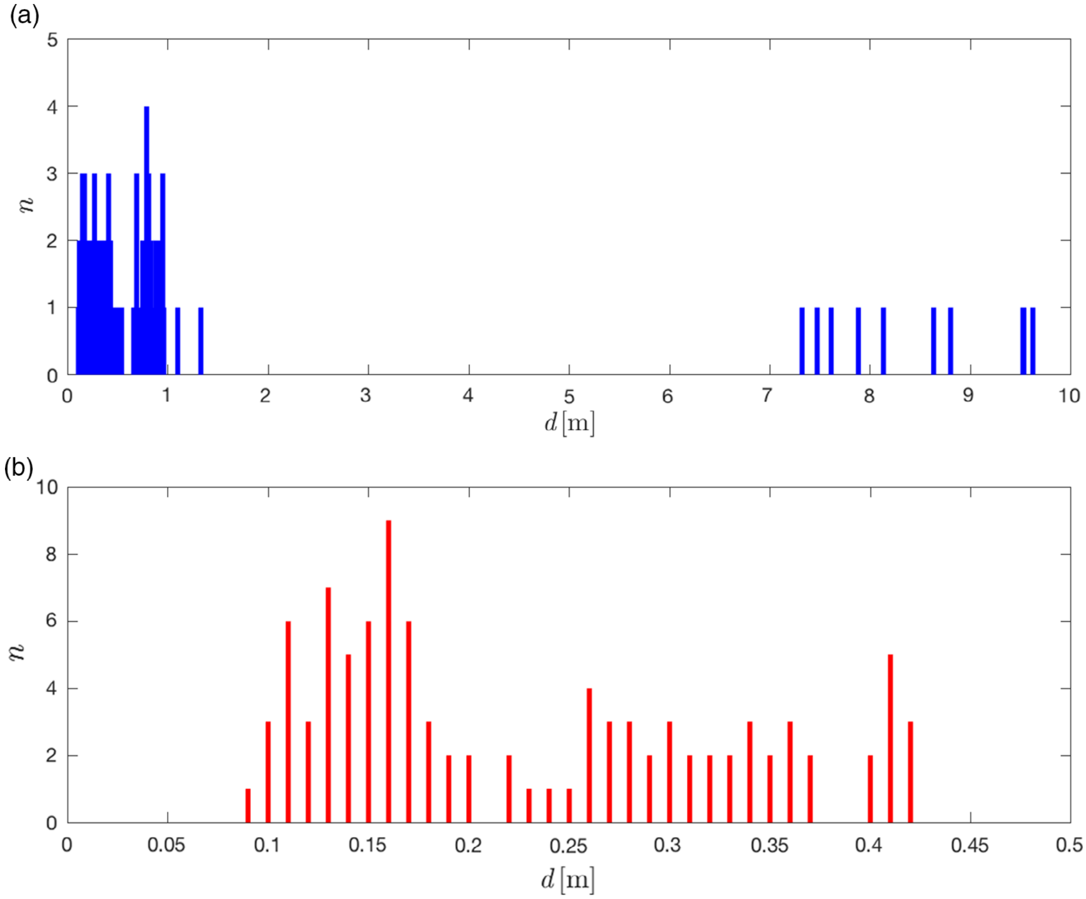

Figure 8. Distribution of the values of $d^{(j)}$ when the TN is positioned in $\underline{u} = [-3, 7]$ and when considering the distance-based algorithm (upper figure) and the improved algorithm (lower figure).

$\{d^{(j)}\}_{j=1}^{100}$ are lower than 1 m, but in some cases the distance error can increase to almost 10 m. On the contrary, when considering the improved algorithm the values of $\{d^{(j)}\}_{j=1}^{100}$ are always smaller than 0.5 m. These results are in agreement with those shown in Figure 4, where it is shown that the value of the average distance error corresponding to the TN positioned in $\underline{u} = [-3, 7]$ considerably reduces when using the improved version of the localization algorithm.

The previous examples show how the values of $\{\text{vPeak}_i\}_{i=0}^{2}$ used by the improved algorithm, i.e., accurate estimates of the LoS/NLoS status of a link, are expedient to improve the localization accuracy.

6. Conclusion

The performance of a UWB-based localization system in a realistic industrial environment has been investigated. The scenario of interest involves the use of three UWB modules that are installed on a LGV and act as ANs. An additional UWB module is used as TN and its position is determined using multiple range estimates from the ANs. We have proposed two localization algorithms: the first one only uses range estimates to perform localization, whereas the second one also uses the values of vPeak corresponding to improved RSS. Our results show that the proposed localization method, especially the hybrid distance/vPeak improved one, allows obtaining accurate position estimates of the TN in most of the positions around the LGV, despite NLoS phenomena. This makes its applicability attractive.

As a matter of fact, our final aim would be to use the proposed localization approach as a collision avoidance system, to improve safety inside indoor industrial environments, and, in particular, to avoid possible accidents between LGVs and people/manual vehicles inside warehouses. Considering a static TN, as done in this article, makes the scenario simpler than a realistic one. However, the experimental campaign that led to the results shown in this article should be considered as a useful and necessary step to evaluate the performance of the proposed algorithms to estimate the position of TNs in various positions around the LGV.

According to the obtained results, the localization accuracy of the localization algorithm based on distance estimates and on vPeak is good in most of the regions around the LGV, especially in the area in front of the LGV, i.e., the area toward which the LGV would move in a realistic scenario. These results are particularly promising, as our final aim is to improve safety in a realistic scenario where the LGV is moving forward.

As a future development in this research, the accuracy of the proposed localization algorithms will be analyzed in a dynamic scenario where the LGV and/or the TN move inside a warehouse. According to our preliminary results, the proposed localization method guarantees to obtain accurate position estimates also in dynamic scenarios, thanks to the high temporal resolution of acquired data. In the envisaged application scenario, LGVs coexist with manual forklifts and people, which can be considered as TNs. Each LGV has its own reference system and can locate TNs independently of other LGVs, possibly relying also on data acquired from other wireless technologies, such as WiFi, and on data acquired from the sensors available on the LGV, such as the (known) speed of the LGV. Although indoor localization has been extensively studied for different aims, such as micro aerial vehicles,[30,31] we are not aware of any commercial solution ready to be used for this purpose.

Conflict of Interest

The authors declare no conflict of interest.

Keywords

indoor localizations, laser-guided vehicles, ultra wide bands

Received: April 21, 2020
Revised: October 25, 2020
Published online: December 3, 2020

[1] Z. Sahinoglu, S. Gezici, I. Guvenc, *Ultra-Wideband Positioning Systems: Theoretical Limits, Ranging Algorithms and Protocols*, Cambridge University Press, Cambridge, U.K. **2008**.
[2] D. Cassioli, M. Z. Win, A. F. Molisch, in *IEEE Vehicular Technology Conf.*, IEEE **2001**, pp. 1159–1163.
[3] S. Gezici, H. V. Poor, *Proc. IEEE* **2009**, *97*, 386.
[4] D. Dardari, C. C. Chong, M. Z. Win, *IEEE Trans. Commun.* **2008**, *56*, 1366.
[5] S. Gezici, T. Zhi, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, Z. Sahinoglu, *IEEE Signal Process. Mag.* **2005**, *22*, 70.
[6] S. A. Mitilineos, A. D. Nick, M. T. Effie, K. M. Dimitris, S. C. A. Thomopoulos, in *Proc. of the 6th Int. Conf. on Advances in Mobile Computing and Multimedia*, Association for Computing Machinery, New York, NY **2008**, pp. 178–182.
[7] E. Sansano, O. Belmonte-Fernández, R. Montoliu, A. Gascó-Compte, A. Caballer Miedes and P. Bayarri Iturralde, in *2019 15th Int. Conf. on Intelligent Environments*, IEEE **2019**, pp. 22–29.
[8] X. Huang, E. Dutkiewicz, R. Gandia, D. Lowe, in *IEEE Int. Conf. on Industrial Informatics*, IEEE **2006**, pp. 1012–1017.
[9] R. Chávez-Santiago, I. Balasingham, J. Bergsland, *J. Electr. Comput. Eng.* **2012**, *3*, 1.
[10] A. Chehri, H. T. Mouftah, *Concurr. Comput. Pract. Exp.* **2020**, *32*, 2.
[11] J. Romme, B. Kull, in *Proc. of the IEEE Int. Symp. on Consumer Electronics*, IEEE **2002**.
[12] F. Khelifi, A. Bradai, A. Benslimane, P. Rawat, M. Atri, *Mobile Netw. Appl.* **2018**.
[13] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, N. S. Correal, *IEEE Signal Process. Mag.* **2005**, *22*, 54.
[14] V. Barral, P. Suárez-Casal, C. J. Escudero, J. A. García-Naya, *Electronics* **2019**, *8*, 1152.
[15] J. Kim, H. Cho, S. Kim, *Int. J. Fuzzy Logic Intell. Syst.* **2013**, *13*, 307.
[16] X. Liao, R. Chen, M. Li, B. Guo, X. Niu, W. Zhang, *Sensors* **2019**, *19*, 11261.
[17] J. Li, C. Wang, X. Kang, Q. Zhao, *Int. J. Digital Earth* **2019**, *1*.
[18] Q. Niu, M. Li, S. He, C. Gao, S. H. Gary Chan, X. Luo, *ACM Trans. Sensor Networks* **2019**, *15*, 19.
[19] Y. T. Wang, J. Li, R. Zheng, D. Zhao, in *Proc. IEEE Int. Conf. on Indoor Positioning and Indoor Navigation*, IEEE **2017**, pp. 1–8.
[20] K. Khoshelham, S. O. Elberink, *Sensors* **2012**, *12*, 1437.
[21] D. Vandermeulen, C. Vercauteren, M. Weyn, in *Proc. Int. Conf. on Ambient Computing, Applications, Services and Technologies* **2013**, pp. 42–49.

[22] Y. Wu, H. Zhu, Q. Du, *Int. J. Automat. Comput.* **2019**, *16*, 1437.

[23] W. Shi, J. Du, X. Cao, Y. Yu, Y. Cao, S. Yan, C. Ni, *Sensors* **2019**, *19*, 968.

[24] D. Vasisht, S. Kumar, D. Katabi, in *Symp. on Networked Systems Design and Implementation,*USENIX Association, United States **2016**, pp. 165–178.

[25] A. Karaagac, J. Haxhibeqiri, M. Ridolfi, W. Joseph, I. Moerman, J. Hoebeke, in *22nd IEEE Int. Conf. on Emerging Technologies and Factory Automation,* IEEE **2017** pp. 1–8.

[26] http://www.ubisense.net/en (accessed: February, 2020).

[27] http://www.timedomain.com (accessed: February, 2020).

[28] Time Domain, Ranging and communications application programming interface (API) specification version 2.1 **2012**, 320-0282E.

[29] S. Monica, G. Ferrari, in *Proc. of the IEEE Int. Conf. on Ultra Wide Band*, IEEE **2014**, pp. 297–302.

[30] K. Guo, Z. Qiu, W. Meng, L. Xie, R. Teo, *Int. J. Micro Air Vehicles* **2017**, 169.

[31] S. van der Helm, K. N. McGuire, M. Coppola, G. C. H. E. de Croon, *Autonom. Robot* **2018**, 1.