

Gaussian Message Passing-Based Cooperative Localisation with Bootstrap Percolation Scheme in Dense Networks

Yangyang Liu, Baowang Lian, Lin Zhang and Taoyun Zhou

*(School of Electronics and Information, Northwestern Polytechnical University,
Xi'an 710072, China)*

(E-mail: liu_yang90@163.com)

In dense networks, the tremendous computational complexity and communication overhead of cooperative localisation are the two main bottlenecks that limit practical application. In this study, we introduce a bootstrap percolation scheme into Gaussian message passing-based cooperative localisation for precise positioning, aimed at reducing the system overhead. Considering the uncertainty information and geometric distribution of neighbours, an approximate collinear detection criterion is proposed to detect the possible flip ambiguities in cooperative localisation. According to the detection result and our connection constraint, agents are divided into three categories and are approximated by different distribution families. A message passing rule is designed to control the propagation direction from high precision to low precision, thereby mitigating potential error propagation. Additionally, a layer-by-layer positioning mechanism is established where the agents are located gradually. Analytical and simulation results indicate that when the ranging standard deviation is 0.2 m, 89.3% of the agent nodes can be located within 0.4 m using the proposed algorithm. Compared with the Hybrid Sum-Product Algorithm over A Wireless Network (H-SPAWN), this ratio is increased by 8.4% and the computational complexity is reduced by 72%.

KEY WORDS

1. Cooperative localization.
2. Bootstrap percolation scheme.
3. Approximate collinear detection.
4. Flip ambiguities.

Submitted: 25 January 2018. Accepted: 7 March 2019. First published online: 2 May 2019.

1. INTRODUCTION. Location awareness has become a fundamental service in Wireless Sensor Networks (WSNs), including military surveillance, smart city, precision agriculture and the Internet of Things (Abouzar et al., 2016; Wang et al., 2015; Rantakokko et al., 2011). However, precise positioning is still a great challenge in indoor, tunnel and other harsh environments due to the lack of Global Navigation Satellite System (GNSS) signals (Fan et al., 2017). In challenging environments, Ultra-Wide Bandwidth (UWB)-based localisation technology has attracted great interest due to its precise ranging, simple implementation for multiple-access communications and high reliability (Wymeersch et al., 2009). UWB technology uses Time-Of-Arrival (TOA)-based ranging techniques with a

fine delay resolution due to its wideband characteristics. Typically, the UWB-based ranging accuracy can reach 0.2 m in a line-of-sight environment, which provides a foundation for high-precision positioning in wireless networks (Caceres et al., 2011; Dardari et al., 2009).

Cooperative localisation is a promising way to improve both the coverage and accuracy of a positioning system by exploiting peer-to-peer communication and ranging (Li et al., 2015b). In a Bayesian cooperative localisation algorithm, the agent node is treated as a random variable and its full statistical information is exchanged and updated between neighbours (Ihler et al., 2005). As a novel cooperative localisation method, a message passing algorithm maps the Bayesian inference problem to the probabilistic graphical model and employs a spatiotemporal message schedule to estimate the agent's marginal distribution by message propagation, computation and updating (Lv et al., 2016; Wymeersch et al., 2009).

In particular, a Sum-Product Algorithm over A Wireless Network (SPAWN) is a sample-based distributed message passing algorithm which exchanges full statistical information between neighbours, resulting in significant system overhead (Wymeersch et al., 2009). Based on the SPAWN, the H-SPAWN (Hybrid SPAWN) algorithm (Caceres et al., 2011; Li et al., 2015a), as a typical parametric message passing method, exploits the Gaussian distribution to approximate the irregular messages and only a few Gaussian parameters need to be exchanged between agents. Gaussian parametric message approximation is also employed in Variational Message Passing (VMP) to reduce system overhead by exploiting second-order Taylor expansions (Cui et al., 2017). Sigma-Point Belief Propagation (SPBP) extends the sigma point filter to general factor structures and reformulates the messages in higher-dimensional spaces, just broadcasting a mean vector and a covariance matrix between neighbours (Meyer et al., 2014; Georges et al., 2016). SPBP is suitable for nonlinear systems, but not for dense networks, because the dimension and the number of sigma points depend on the number of neighbours (Meyer et al., 2014). H-SPAWN, VMP and SPBP are all devoted to alleviating the system overhead of single trial message passing and computation. However, in each iteration of these methods, all the agent nodes need to estimate their position based on all messages from neighbours, even though the contributions of some messages from neighbours may be neglected.

In dense networks, the excessive number of neighbours leads to a huge computation burden, but also may bring an adverse effect on accuracy (Jing et al., 2016; Das and Wymeersch, 2012; Velde et al., 2015). In order to further reduce the system overhead in dense networks, some neighbour node selection schemes, also known as information censoring, have been proposed to attempt to refine the most informative neighbour subset for positioning, rather than all neighbour messages in the above-mentioned methods. Inspired by Global Navigation Satellite Systems (GNSS), Das and Wymeersch (2012) proposed the Modified Bayesian Fisher Information Matrix (MBFIM), which combines Geometric Dilution Of Precision (GDOP) with the target's prior information as the metric to remove the unreliable neighbour information and applies this node selection scheme to SPAWN. In contrast with Das and Wymeersch (2012), the factor of a neighbour's uncertainty is considered by Velde et al. (2015) in the Greedy Censoring Scheme (GCS), which proposes a Bayesian Cramer-Rao Lower Bound (BCRLB) censoring metric to predict the average position accuracy and this scheme is applied to SPBP.

In addition, the bootstrap percolation scheme has brought a new perspective to cooperative localisation due to its hierarchical gradual positioning (Lv et al., 2016; Vaze and Gupta,

2012; Cai et al., 2014). The bootstrap percolation is introduced to iterative localisation to solve the problem of the minimum number of anchor nodes under the condition that the whole network is localised (Vaze and Gupta, 2012). In order to control the error propagation in non-Bayesian cooperative localisation, a low-complexity linear least squares estimator is designed in Cai et al. (2014) by using the bootstrap percolation with soft constraints. In combining the bootstrap percolation with Non-parametric Belief Propagation (NBP) Lv et al. (2016) proposed a space-time hierarchical-graph-based cooperative localisation algorithm to reduce system overhead and control error propagation.

However, the direct use of a bootstrap percolation scheme without judgement may bring a catastrophic problem to cooperative localisation. In cooperative localisation, positioning error is related to the ranging error, the geometric configuration of reference nodes and their uncertainties. A line will be formed when the reference nodes are approximately collinear. In this case, target position estimation may be reflected on the other side of the line, thereby causing an unbearable positioning error. This phenomenon is called “flip ambiguity” (Kannan et al., 2010; Han et al., 2015; Wang et al., 2013). What is more, the flip positioning result may propagate to the subsequent agent nodes, which may cause an “avalanche” error propagation in the subsequent iterations, resulting in a failure to locate a large portion of the agents.

In this paper, we focus on introducing the bootstrap percolation scheme into message-passing-based cooperative localisation in dense WSNs. In order to eliminate the potential flip ambiguities, an approximate collinear detection criterion considering the geometric configuration and uncertainties of the neighbours is proposed, and then the quality of the neighbour reference nodes is evaluated. Combining the detection results with the connection constraints, the agent nodes are divided into three categories, which are respectively approximated by the ring distribution, the Gaussian mixture distribution¹ and the Gaussian distribution. Then, a layer-by-layer message propagation and updating mechanism is designed to control the information propagation and mitigate the potential error propagation. Additionally, the Gaussian parametric message passing method is proposed to reduce the communication overhead and facilitate efficient message computation. Finally, the system overhead and performance are evaluated.

2. SYSTEM MODEL. In WSNs, a small number of sensors with known positions are referred to as anchors, while the others which need to be located by employing the ranging information between their own neighbours are referred to as agents. Figure 1 shows a typical application of cooperative localisation in a road environment. In Figure 1, each agent cannot independently estimate its position based on the ranging information associated with the anchor nodes. In cooperative localisation, the agent can determine its own position by exploiting peer to peer ranging and communication between neighbours. Therefore, compared with non-cooperative localisation, cooperative localisation can significantly improve the accuracy and coverage of the positioning.

We consider a two-dimensional WSN consisting of N anchor nodes and M agent nodes, which can be denoted by sets \mathcal{N} and \mathcal{M} , respectively. We assume that all the nodes are UWB sensors because of their peer-to-peer communication and relative distance measurement capabilities. The position of the anchors is known exactly, and the agents are randomly

¹ Note that the Gaussian mixture distribution refers to the Gaussian mixture distribution with two components in this work.

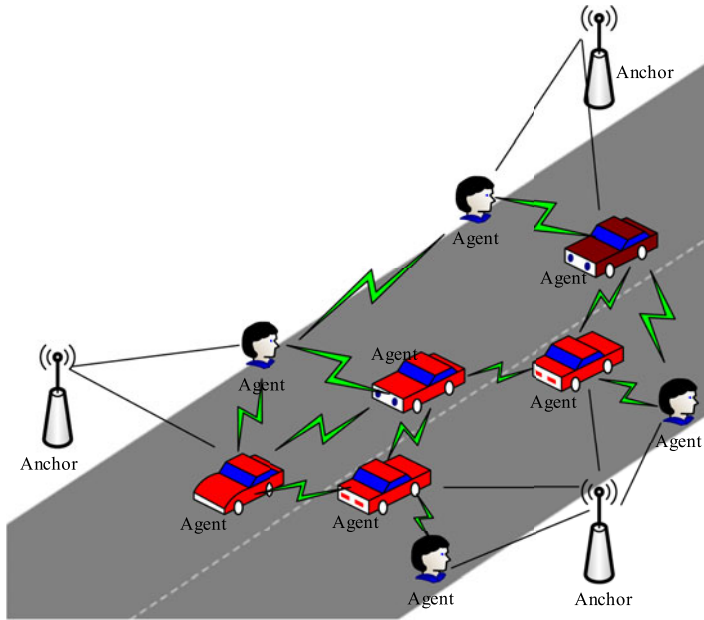


Figure 1. Diagram of cooperative localisation in a road environment.

scattered in the cooperative localisation area. Let $\mathbf{x}_i = [x_i \ y_i]^T$ be the position of node i . The agent set \mathbb{M} is divided into two subsets: the agents which have been located belong to set \mathbb{C} and the others still with unknown positions make up the set \mathbb{A} . In cooperative localisation, the located agents in set \mathbb{C} have relatively small uncertainties, which can be used to assist the agents in set \mathbb{A} . The reference node set is defined as $\mathbb{S} = \mathbb{N} \cup \mathbb{C}$. The reference nodes broadcast their own position information, while the agents belonging to set \mathbb{A} remain silent. Each node can range and communicate with neighbours within the maximum sense distance R . We define notation \mathbb{S}_i as the reference node set received by agent i . For each node $j \in \mathbb{S}_i$, the measurement distance from node j to node i is perturbed by measurement noise as:

$$z_{j \rightarrow i} = \|\mathbf{x}_j - \mathbf{x}_i\| + v_{j \rightarrow i} \tag{1}$$

where $\|\cdot\|$ is the Euclidean norm and $v_{j \rightarrow i} \sim \mathcal{N}(0, \sigma_{j \rightarrow i}^2)$ is the additive white Gaussian measurement noise in the line of sight.

We define the notation $\mathbb{Z}_i = \{z_{j \rightarrow i} | \forall j \in \mathbb{S}_i\}$ as the range measurements set for node i . The belief of agent node i , which serves as an approximation of the marginal *a posteriori* distribution $p(\mathbf{x}_i | \mathbb{Z}_i)$, is proportional to its prior distribution $p(\mathbf{x}_i)$ and to the product of the received messages corresponding to its reference nodes. Then the belief is given as:

$$b(\mathbf{x}_i) \propto p(\mathbf{x}_i) \prod_{j \in \mathbb{S}_i} m_{f_{i,j} \rightarrow i}(\mathbf{x}_i) \tag{2}$$

where belief $b(\mathbf{x}_i)$ is an *a posteriori distribution* estimation of node i . $m_{f_{i,j} \rightarrow i}(\mathbf{x}_i)$ is the message from node j to node i , which consists of three parts: the likelihood function $p(z_{j \rightarrow i} | \mathbf{x}_i, \mathbf{x}_j)$ associated with the ranging measurement between node i and j , the prior

distribution of node j , and the messages merging into node j . According to the message updating rule, $m_{f_{j \rightarrow i}}(\mathbf{x}_i)$ is calculated by:

$$\begin{aligned}
 m_{f_{j \rightarrow i}}(\mathbf{x}_i) &\propto \int p(z_{j \rightarrow i} | \mathbf{x}_i, \mathbf{x}_j) p(\mathbf{x}_j) \prod_{u \in \mathbb{S}_j} m_{f_{j,u \rightarrow j}}(\mathbf{x}_j) d\mathbf{x}_j \\
 &\propto \int p(z_{j \rightarrow i} | \mathbf{x}_i, \mathbf{x}_j) b(\mathbf{x}_j) d\mathbf{x}_j
 \end{aligned}
 \tag{3}$$

3. PROPOSED MESSAGE PASSING ALGORITHM. In this section, we present the proposed message passing-based cooperative localisation algorithm. It is well known that the target should be connected to at least three non-collinear reference nodes to obtain a two-dimensional position estimation. Based on this consensus, the idea of bootstrap percolation subject to connection and geometric constraints is exploited to classify the agent nodes. The connection constraint guarantees the number of reference nodes, and the geometric constraint ensures that the reference nodes are not approximately collinear. Considering the measurement errors, the approximate collinear reference nodes may lead to a flip ambiguity of the target position estimation. In order to eliminate the potential flip ambiguities, we propose an approximate collinear detection criterion to satisfy the geometric constraint. Then, a layer-by-layer positioning mechanism is established associated with the classification result. Finally, the Gaussian message passing localisation algorithm is employed to locate the agents gradually, based on the proposed bootstrap percolation scheme.

3.1. *Message representation method.* In distributed cooperative localisation, “message” refers to the node’s position estimation and its uncertainty. In this work, the Gaussian distribution is used to approximate the message of the reference node, and only five real numbers (two are the mean vector and the other three are from the covariance matrix) need to be broadcast for message passing.

Without loss of generality, for reference node $j \in \mathbb{S}_i$, we assume that $b(\mathbf{x}_j) = \mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j)$, where $\boldsymbol{\mu}_j$ is the mean and Σ_j is the covariance matrix. The distance measurement from node j to node i is $z_{j \rightarrow i}$, but the information regarding the direction is unknown. Hence, the distribution of node i can be described as a ring distribution with a centre of \mathbf{x}_j and a radius of r , which can be further expressed as:

$$\mathbf{x}_i = \mathbf{x}_j + r \times [\cos \theta \ \sin \theta]^T
 \tag{4}$$

where $\mathbf{x}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j)$, $r \sim \mathcal{N}(z_{j \rightarrow i}, \sigma_{j \rightarrow i}^2)$, and $\theta \sim U(0, 2\pi]$. It should be noted that if node j is an anchor node, then $\Sigma_j = 0$.

As shown in Figure 2, we assume $\hat{\mathbf{x}}_i$ is the estimation of the node i , and $\hat{\theta} = \text{angle}(\hat{\mathbf{x}}_i - \boldsymbol{\mu}_j)$ is the direction between node i and node j . In order to obtain the variance $\sigma_{j, \hat{\theta}}^2$ associated with the marginal distribution of $b(\mathbf{x}_j)$ along $\hat{\theta}$, the coordinate system is rotated anticlockwise by angle $\hat{\theta}$. Then the covariance matrix of \mathbf{x}_j in the new coordinate system is given as:

$$\Sigma_{j, \hat{\theta}} = H_{\hat{\theta}}^T \Sigma_j H_{\hat{\theta}}
 \tag{5}$$

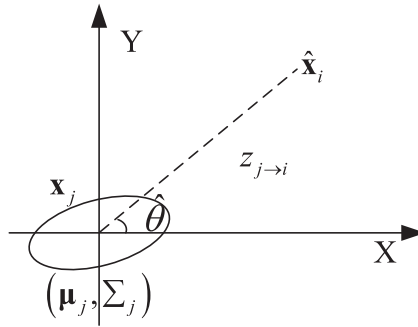


Figure 2. Variance estimation of Gaussian marginal distribution.

where:

$$H_{\hat{\theta}} = \begin{bmatrix} \cos(\hat{\theta}) & -\sin(\hat{\theta}) \\ \sin(\hat{\theta}) & \cos(\hat{\theta}) \end{bmatrix} \tag{6}$$

Hence, the $\sigma_{j,\hat{\theta}}^2$ associated with the (1,1)-th element of $\Sigma_{j,\hat{\theta}}$ is shown as:

$$\sigma_{j,\hat{\theta}}^2 = [\cos(\hat{\theta}), \sin(\hat{\theta})] \Sigma_j [\cos(\hat{\theta}), \sin(\hat{\theta})]^T \tag{7}$$

The marginal distribution of \mathbf{x}_j along any direction is still a Gaussian distribution, thus, we can rewrite Equation (4) as:

$$\mathbf{x}_i = \mu_j + r_{\theta} \times [\cos \theta \ \sin \theta]^T \tag{8}$$

where $r_{\theta} \sim \mathcal{N}(z_{j \rightarrow i}, \sigma_{j \rightarrow i, \theta}^2)$ and $\sigma_{j \rightarrow i, \theta}^2 = \sigma_{j \rightarrow i}^2 + \sigma_{j, \theta}^2$.

Hence, the joint *a posteriori* distribution of \mathbf{x}_i is related to the number of received neighbour nodes $|\mathbb{S}_i|$, which can further be considered as the overlapping areas of $|\mathbb{S}_i|$ rings. When $|\mathbb{S}_i| = 2$, the intersection area of the two rings is the estimation of the target’s position. This is a bimodal distribution and we employ a Gaussian mixture distribution to approximate it. For $|\mathbb{S}_i| \geq 3$, a steady unimodal distribution is obtained when the number of non-collinear reference nodes is no less than three and we approximate it with a Gaussian distribution.

Without loss of generality, the equation set of observation between the target and its neighbour reference nodes can be formulated as follows:

$$\begin{cases} (x_1 - x_i)^2 + (y_1 - y_i)^2 = z_{1 \rightarrow i}^2 \\ \vdots \\ (x_n - x_i)^2 + (y_n - y_i)^2 = z_{n \rightarrow i}^2 \end{cases} \tag{9}$$

where $n = |\mathbb{S}_i|$. We define notation $\hat{\mathbf{x}}_i^a$ and $\hat{\mathbf{x}}_i^b$ as the solutions of Equation (9) when $|\mathbb{S}_i| = 2$, and the notation $\hat{\mathbf{x}}_i$ for $|\mathbb{S}_i| \geq 3$. Then the approximate distribution can be shown as follows:

$$p(\mathbf{x}_i | \mathbb{Z}_i) = \begin{cases} C(\mathbf{x}_i; \mathbf{x}_j, z_{j \rightarrow i}, \sigma_{j \rightarrow i}^2), & |\mathbb{S}_i| = 1 \\ \frac{1}{2} C_J(\mathbf{x}_i; \hat{\mathbf{x}}_i^a, \sigma_i^2 \mathbf{I}) + \frac{1}{2} C_J(\mathbf{x}_i; \hat{\mathbf{x}}_i^b, \sigma_i^2 \mathbf{I}), & |\mathbb{S}_i| = 2 \\ C_J(\mathbf{x}_i; \hat{\mathbf{x}}_i, \sigma_i^2 \mathbf{I}), & |\mathbb{S}_i| \geq 3 \end{cases} \tag{10}$$

where the notations C and C_j refer to the ring and Gaussian distributions, respectively. In the ring distribution, parameter \mathbf{x}_j is the centre of the ring and $z_{j \rightarrow i}$ is the radius with a variance $\sigma_{j \rightarrow i}^2$. In the Gaussian distribution, parameter $\hat{\mathbf{x}}_i$ is the mean, $\sigma_i^2 \mathbf{I}$ is the variance matrix, and \mathbf{I} is the identity matrix. For simplicity, let $\sigma_i^2 = \max \left\{ \sigma_{j \rightarrow i, \theta}^2 \mid \forall j \in \mathbb{S}_i \right\}$.

According to the above analysis, it can be seen that the distribution of the target belongs to the different families in terms of the number of the received reference nodes $|\mathbb{S}_i|$. Therefore, for agent i , the message can be represented by randomly generating k samples $\{\mathbf{x}_i(n), w_i(n)\}_{n=1}^k$ according to Equation (10). As an example, Figure 3 shows the generated samples associated with different distribution families.

3.2. *Approximate collinear detection criterion.* There should be at least three reference nodes for a target node to achieve a two-dimensional position estimation, but this is a necessary condition rather than a sufficient condition. Together with the existence of ranging errors, an approximate collinear geometric configuration may cause a flip ambiguity for the target. An illustration of flip ambiguity is shown in Figure 4, where the reference nodes A, B and C are approximately collinear. Hence, the position of target O may be incorrectly estimated at O' , where O and O' are symmetrical about line AB . This flip ambiguity may propagate to the subsequent target nodes and lead to a fatal impact on the overall network.

In order to avoid the flip ambiguity, we propose an approximate collinear detection criterion to judge the geometric configuration of the reference nodes. As shown in Figure 5, we assume that the reference node set of target node i is $\mathbb{S}_i = \{A, B, C, D, E, F\}$. Without considering the covariance of the reference nodes, the distance between any two reference nodes is calculated and the line with the maximum distance (for example, AB) is regarded as the boundary. Then the remaining reference nodes are divided into two subsets, which are $\mathbb{S}_{i1} = \{C, D\}$ and $\mathbb{S}_{i2} = \{E, F\}$, respectively. We suppose that the reference nodes C and E have the maximum vertical distance to line AB in each's own subset, and their vertical distances are d_{i1} and d_{i2} , respectively. The reference node set $\{A, B, C, E\}$ contains the main geometric characteristic of the \mathbb{S}_i . Therefore, we just need to perform the approximate collinear detection on $\{A, B, C, E\}$, rather than \mathbb{S}_i .

For the target node, its received message from the reference node is approximated by $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$. Here we take the reference node A in Figure 5 to further illustrate our detection criteria. We define notation r as the Mahalanobis distance (De Maesschalck et al., 2000) of a point $\mathbf{x} = [x, y]^T$ to the reference node A , which is given as:

$$r^2 = (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \tag{11}$$

It is well known that the locus of the point \mathbf{x} is an ellipse when r is fixed. It is assumed that the probability of a generated sample from the Gaussian distribution of the reference node A falling into the ellipse with a certain Mahalanobis distance r is p , then the relationship between r and p is deduced as follows:

$$r = \sqrt{-2 \ln(1 - p)} \tag{12}$$

We define notation ζ_A as the semi-major axis of the ellipse with a certain r , and notations λ_1 and λ_2 as the two eigenvalues of the covariance matrix Σ associated with reference node A . If $\lambda_1 > \lambda_2$, we have:

$$\zeta_A = r \sqrt{\lambda_1} \tag{13}$$

The detailed derivation is presented in (Bensimhou, 2009).

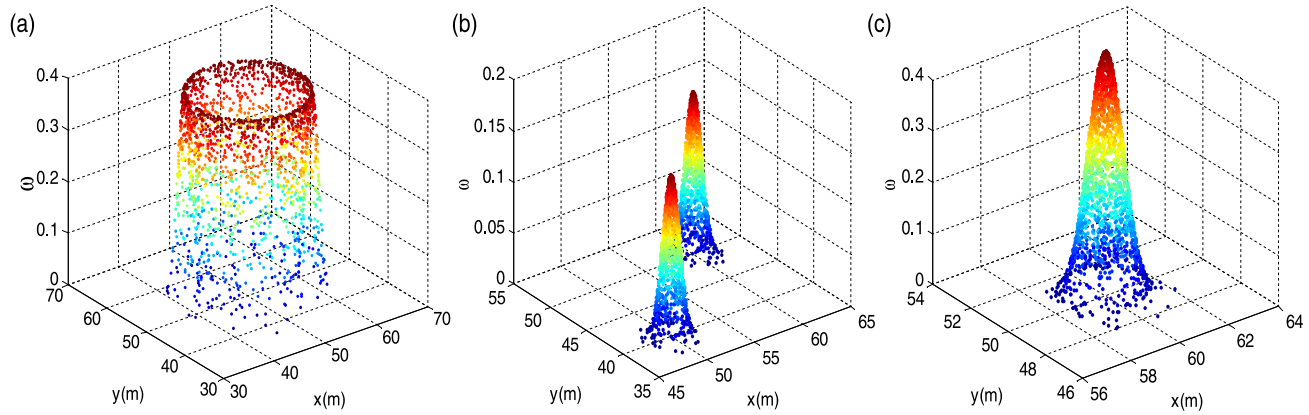


Figure 3. Examples of generated samples from different distribution families for agent nodes. (a) $|\mathbb{S}_i| = 1$, the ring distribution; (b) $|\mathbb{S}_i| = 2$, the Gaussian mixture distribution; (c) $|\mathbb{S}_i| \geq 3$, Gaussian distribution.

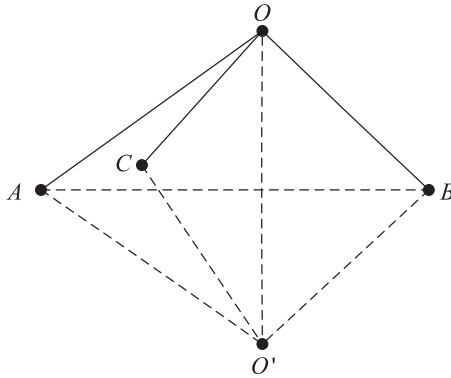


Figure 4. An illustration of flip ambiguity.

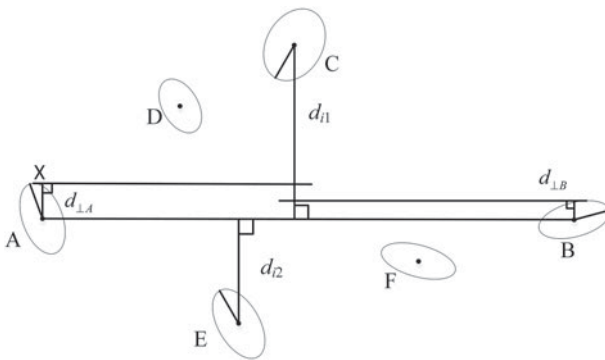


Figure 5. The geometric configuration of the reference node set for a target. The uncertainty ellipses represent the confidence interval of probability p .

As shown in Figure 5, for reference node A , $d_{\perp A}$, the maximum vertical distance of a point X on the ellipse to the line AB , is satisfied as:

$$d_{\perp A} \leq d_{AX} \leq \zeta_A \tag{14}$$

where d_{AX} is the distance from point X to the ellipse centre. It is obtained that $d_{\perp B} \leq \zeta_B$ for reference node B in the same way. Then it can be shown that:

$$d_{\perp m} = \max \{d_{\perp A}, d_{\perp B}\} \leq \max \{\zeta_A, \zeta_B\} = \zeta_m \tag{15}$$

Without loss of generality, it is assumed that $d_{i1} \geq d_{i2}$ as shown in Figure 5, and a threshold $\bar{\epsilon} > 0$ is predefined. Then, the approximate collinear detection criterion is shown as:

$$\begin{cases} d_{i1} + d_{i2} - \zeta_C - \zeta_E - 2\zeta_m > \bar{\epsilon} & \text{s.t. } d_{i2} \geq \zeta_m + \zeta_E \\ d_{i1} - \zeta_C - \zeta_m > \bar{\epsilon} & \text{s.t. } d_{i2} < \zeta_m + \zeta_E \end{cases} \tag{16}$$

Hence, it is considered that the reference nodes are not approximately collinear when their geometric configuration satisfies the constraint of Equation (16), otherwise, it may lead to flip ambiguity in the network.

3.3. *Bootstrap percolation scheme.* In this subsection, the bootstrap percolation scheme is introduced into message passing-based cooperative localisation to control the information propagation between neighbours. We define notation \mathbb{H}^l as the set of agent nodes which are located in the l -th iteration. Then, the set of located agent nodes given by the previous l iterations is achieved as:

$$\mathcal{C}^l = \bigcup_m \mathbb{H}^m, m = 1, 2, \dots, l - 1 \tag{17}$$

Therefore, in the l -th iteration, the set of reference nodes is $\mathcal{S}^l = \mathcal{N} \cup \mathcal{C}^l$, and the set of the remaining agents with unknown position is $\mathcal{A}^l = \mathcal{M} \setminus \mathcal{C}^l$. Here, we define the piecewise function $T_f(\cdot)$ which is associated with the number of reference nodes of the target i as:

$$T_f(|\mathcal{S}_i|) = \begin{cases} 1, & |\mathcal{S}_i| = 1, \\ 2, & |\mathcal{S}_i| = 2, \\ 3, & |\mathcal{S}_i| \geq 3, \end{cases} \tag{18}$$

Then, \mathbb{A}^l is divided into three subsets, which are denoted as:

$$\mathbb{A}_t^l = \{i | i \in \mathbb{A}^l, T_f(|\mathcal{S}_i^l|) = t\} \tag{19}$$

where $t = 1, 2, 3$. As analysed in Section 3.1, the distribution of the target i is a ring when $i \in \mathbb{A}_1^l$, and it is a Gaussian mixture distribution in the case of $i \in \mathbb{A}_2^l$. However, for $i \in \mathbb{A}_3^l$, according to the approximate collinear detection result of \mathcal{S}_i , it may have two different types of distribution: the Gaussian mixture distribution and the Gaussian distribution. Therefore, for each target $i \in \mathbb{A}_3^l$, we assume $i \in \mathbb{A}_{3+}^l$ if it satisfies Equation (16), otherwise, we assume $i \in \mathbb{A}_{3-}^l$. Combining the approximate collinear detection result with the function $T_f(\cdot)$, we re-divide \mathbb{A}^l as:

$$\mathbb{B}_c^l = \begin{cases} \mathbb{A}_1^l, & c = 1, \\ \mathbb{A}_2^l \cup \mathbb{A}_{3-}^l, & c = 2, \\ \mathbb{A}_{3+}^l, & c = 3, \end{cases} \tag{20}$$

where c is the confidence threshold and $c = 1, 2, 3$.

It can be seen from Equation (20) that the agent nodes belonging to \mathbb{B}_3^l have the highest confidence and can be located without flip ambiguities. Thus, $\mathbb{H}^l = \mathbb{B}_3^l$ when $\mathbb{B}_3^l \neq \emptyset$, while the agent nodes belonging to \mathbb{B}_1^l and \mathbb{B}_2^l just update their distribution. As the iteration proceeds, \mathbb{B}_2^l will replace \mathbb{B}_3^l to be located when $\mathbb{B}_3^l = \emptyset$, and $\mathbb{H}^l = \mathbb{B}_2^l$ is achieved. To this end, all the agent nodes will be located in this layer-by-layer manner. According to this scheme, the subset with the highest confidence has the priority to be located, thus, \mathbb{H}^l can be described as:

$$\mathbb{H}^l = \{i | i \in \mathbb{B}_c^l, c = \max\{1, 2, 3\}\}, \quad \text{s.t. } \mathbb{B}_c^l \neq \emptyset \tag{21}$$

The detailed bootstrap percolation scheme is summarised in Algorithm 1.

Algorithm 1 Bootstrap percolation scheme.

1. **Initialisation:** The sets of located agent nodes, the reference nodes and the agent nodes with unknown position are $\mathbb{C}^1 = \emptyset$, $\mathbb{R}^1 = \mathbb{N}$ and $\mathbb{A}^1 = \mathbb{M}$, respectively.
 2. **for** iteration $l = 1$ to L .
 3. Divide \mathbb{A}^l into three subsets \mathbb{A}_1^l , \mathbb{A}_2^l and \mathbb{A}_3^l using Equation (19).
 4. For each node $i \in \mathbb{A}_3^l$, execute approximate collinear detection using Equation (16) and divide \mathbb{A}_3^l into \mathbb{A}_{3-}^l and \mathbb{A}_{3+}^l in terms of the detection result.
 5. Re-divide the \mathbb{A}^l using Equation (20).
 6. Obtain the agent node set \mathbb{H}^l which are located in the l -th iteration using Equation (21).
 7. For each node $i \in \mathbb{H}^l$, estimate its position with Algorithm 2.
 8. Update the sets with $\mathbb{C}^{l+1} = \mathbb{C}^l \cup \mathbb{H}^l$, $\mathbb{R}^{l+1} = \mathbb{R}^l \cup \mathbb{H}^l$, and $\mathbb{A}^{l+1} = \mathbb{A}^l \setminus \mathbb{H}^l$, respectively.
 9. **end for**
-

3.4. *Message computation and updating.* As mentioned above, the reference node set includes two types of nodes: the anchor nodes with exact position and the located agent nodes with a certain uncertainty. According to the message computation rule shown in Equation (3), the message from anchor a to target i is simplified as:

$$M_{f_{i,a} \rightarrow i}(\mathbf{x}_i) \propto p(z_{a \rightarrow i} | \mathbf{x}_i, \mathbf{x}_a) \propto \frac{1}{\sqrt{2\pi\sigma_{a \rightarrow i}^2}} \exp \left\{ -\frac{(z_{a \rightarrow i} - \|\mathbf{x}_a - \mathbf{x}_i\|)^2}{2\sigma_{a \rightarrow i}^2} \right\} \quad (22)$$

where \mathbf{x}_a is the position of anchor a .

For the agent node $j \in \mathbb{S}_i$, the message from node j to target i is expressed as:

$$m_{f_{j \rightarrow i} \rightarrow i}(\mathbf{x}_i) \propto \int \exp \left\{ -\frac{(z_{j \rightarrow i} - \|\mathbf{x}_j - \mathbf{x}_i\|)^2}{2\sigma_{j \rightarrow i}^2} \right\} \mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j) d\mathbf{x}_j \quad (23)$$

It is not possible to compute the Two-Dimensional (2D) integrations in Equation (23) directly because of the nonlinearity of the ranging likelihood function. The nonlinear equation $\|\mathbf{x}_j - \mathbf{x}_i\|$ is linearized by using a first-order Taylor expansion around the position estimations of node i and j , which is shown as:

$$\begin{aligned} \|\mathbf{x}_j - \mathbf{x}_i\| &= \|\hat{\mathbf{x}}_{j \rightarrow i}\| - \frac{\hat{\mathbf{x}}_{j \rightarrow i}}{\|\hat{\mathbf{x}}_{j \rightarrow i}\|} (\mathbf{x}_j - \boldsymbol{\mu}_j) + \frac{\hat{\mathbf{x}}_{j \rightarrow i}}{\|\hat{\mathbf{x}}_{j \rightarrow i}\|} (\mathbf{x}_i - \hat{\mathbf{x}}_i) \\ &= \|\hat{\mathbf{x}}_{j \rightarrow i}\| - \frac{\hat{\mathbf{x}}_{j \rightarrow i}}{\|\hat{\mathbf{x}}_{j \rightarrow i}\|} [(\mathbf{x}_j - \mathbf{x}_i) + \hat{\mathbf{x}}_{j \rightarrow i}] \end{aligned} \quad (24)$$

where $\hat{\mathbf{x}}_{j \rightarrow i} = \hat{\mathbf{x}}_i - \boldsymbol{\mu}_j$. Substituting Equation (24) into Equation (23), Equation (25) may be derived:

$$m_{f_{j \rightarrow i} \rightarrow i}(\mathbf{x}_i) \propto \exp \left[-\frac{1}{2} \left(\hat{\mathbf{x}}_{j \rightarrow i} - \mathbf{z}_{j \rightarrow i} \frac{\hat{\mathbf{x}}_{j \rightarrow i}}{\|\hat{\mathbf{x}}_{j \rightarrow i}\|} \right)^T \Sigma_{j \rightarrow i}^{-1} \left(\hat{\mathbf{x}}_{j \rightarrow i} - \mathbf{z}_{j \rightarrow i} \frac{\hat{\mathbf{x}}_{j \rightarrow i}}{\|\hat{\mathbf{x}}_{j \rightarrow i}\|} \right) \right] \quad (25)$$

where $\Sigma_{j+z} = \Sigma_j + \sigma_{j \rightarrow i}^2 \mathbf{I}$.

According to the message update rule, the weights of the samples for target i can be updated as follows:

$$w_i^l(n) = w_i^l(n) \prod_{j \in \mathbb{S}_i} m_{f_{j \rightarrow i}}(\mathbf{x}_i^l(n)) \tag{26}$$

Normalising the samples so that $\sum_{n=1}^k w_i^l(n) = 1$, we then estimate the mean and covariance of target i as follows:

$$\boldsymbol{\mu}_i = \sum_{n=1}^k w_i(n) \mathbf{x}_i(n) \tag{27}$$

$$\Sigma_i = \frac{\sum_{n=1}^k w_i(n) (\mathbf{x}_i(n) - \boldsymbol{\mu}_i) (\mathbf{x}_i(n) - \boldsymbol{\mu}_i)^T}{1 - \sum_{n=1}^k (w_i(n))^2} \tag{28}$$

In the subsequent iterations, the agent node i is treated as a reference node to broadcast its Gaussian parameters $(\boldsymbol{\mu}_i, \Sigma_i)$ to its neighbours.

Based on the analysis above, the proposed cooperative localisation with bootstrap percolation scheme is described in Algorithm 2.

4. SYSTEM OVERHEAD. In this study, the communication overhead is reduced by: (1) adopting a Gaussian parametric message passing rule and (2) censoring and blocking the agents with unknown position from broadcasting their messages. Thus, the communication overhead of the entire network of the proposed algorithm is $5|\mathbb{C}^l|$ per iteration, while that of H-SPAWN (Caceres et al., 2011) and GCS (Velde et al., 2015) is $5M$, where $|\mathbb{C}^l| \leq M$. For smaller l , this variation is larger. For SPAWN (Wymeersch et al., 2009), it is $3kM$, where k is the number of weighted samples.

Computational complexity is related to the number of samples and the number of messages involved in positioning. The complexity is proportional to the number of samples for the proposed algorithm and H-SPAWN, while for SPAWN, it relies on the square of the number of samples. It should be noted that since the proposed algorithm makes full use of the reference nodes to roughly estimate the target’s distribution, the number of used weighted samples in the proposed algorithm (for example, 200) is much smaller than for SPAWN and H-SPAWN (for example, 2,000). GCS exploits an SPBP algorithm to estimate the target position, and its complexity is cubic in the number of sigma points and, also, in the number of selected neighbour nodes K (Meyer et al., 2014).

The average number of neighbours involved in positioning is exploited to evaluate the influence of the proposed bootstrap percolation scheme on the complexity. We consider a $a \times a$ cooperative localisation area with N anchors and M agents. Assuming that the maximum sense distance between nodes is R , for each target node in SPAWN and H-SPAWN, the average number of links connected to neighbours is:

$$\overline{M}_{SPA} = \frac{N + M}{a^2} \times \pi R^2 = \pi (N + M) \left(\frac{R}{a}\right)^2 \tag{29}$$

Algorithm 2 Proposed cooperative localisation algorithm with bootstrap percolation scheme.

1. **Initialisation:** Set the boundary of the area-of-interest and the prior information of the nodes (for example, Dirac delta function for anchors and random distribution for agents)
 2. For each agent node i , generate k samples $\{\mathbf{x}_i^l(n), w_i^l(n)\}_{n=1}^k$ according to Equation (10).
 3. **for** iteration $l = 1$ to L
 4. **for** each located agent $i \in \mathbb{C}^l$
 5. Compute the message from anchor node using Equation (22).
 6. Compute the message from agent node using Equation (25).
 7. Update the weights using Equation (26).
 8. Estimate the Gaussian parameters $(\boldsymbol{\mu}_i^l, \boldsymbol{\Sigma}_i^l)$ using Equations (27) and (28).
 9. Broadcast $(\boldsymbol{\mu}_i^l, \boldsymbol{\Sigma}_i^l)$ to the neighbours; randomly generate k samples $\{\mathbf{x}_i^{l+1}(n), w_i^{l+1}(n)\}_{n=1}^k$ according to $\mathcal{N}(\boldsymbol{\mu}_i^l, \boldsymbol{\Sigma}_i^l)$ and set $w_i^{l+1}(n) = \exp\left[\frac{1}{2}(\mathbf{x}_i^{l+1} - \boldsymbol{\mu}_i^l)^T (\boldsymbol{\Sigma}_i^l)^{-1} (\mathbf{x}_i^{l+1} - \boldsymbol{\mu}_i^l)\right]$.
 10. **end for**
 11. **for** each agent $i \in \mathbb{A}^l$
 12. **if** receive new message **then**
 13. **if** $i \in \mathbb{H}^l$ **then**
 14. Estimate its position using step 5~9.
 15. **else**
 16. Update its distribution and generate k samples.
 17. **end if**
 18. **else**
 19. Waiting the new message.
 20. **end if**
 21. **end for**
 22. Update the sets in bootstrap percolation scheme.
 23. **end for**
-

GCS uses a fixed number of neighbour nodes K . While for the proposed algorithm, the agents with unknown position remain silent, thus, the average number is:

$$\overline{M}_{PRO} = \frac{N + |\mathbb{C}^l|}{a^2} \times \pi R^2 = \pi (N + |\mathbb{C}^l|) \left(\frac{R}{a}\right)^2 \tag{30}$$

In addition, it should be noted that all the M agents need to be computed in SPAWN, H-SPAWN and GCS, but for the proposed algorithm, only the agents belonging to set $\mathbb{C}^l \cup \mathbb{H}^l$ need to be computed. The detailed communication overhead and computational complexity of the three algorithms are summarised in Table 1.

Table 1. System overhead comparison.

Algorithm	Communication	Complexity	Typical K
Proposed	$5 C^i $	$O((C^i + \mathbb{H}^i) \overline{M}_{PRO}k)$	100~500
SPAWN	$3kM$	$O(M\overline{M}_{SPA}k^2)$	1,000~10,000
H-SPAWN	$5M$	$O(M\overline{M}_{SPA}k)$	1,000~10,000
GCS	$5M$	$O(MK^3k^3)$	15~75

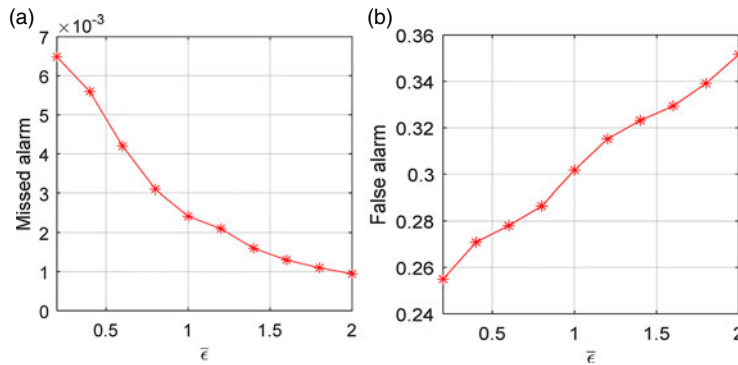


Figure 6. The performance of the approximate collinearity detection criterion, where $\sigma = 0.5$ m and $\bar{\epsilon}$ varies from 0.2 m to 2 m. (a) Missed alarm. (b) False alarm.

5. SIMULATION RESULTS AND DISCUSSIONS.

5.1. *Evaluation of the approximate collinear detection criterion.* In this subsection, the performance of the approximate collinear detection criterion (Equation (16)) is evaluated by numerical simulations. First of all, it is essential to determinate whether the estimated position is a flip ambiguity or not. In Kannan et al. (2010), any position error larger than $R/5$ is treated as a substantial flip ambiguity. This metric is not appropriate when the ranging noise changes. In Han et al. (2015), a flip ambiguity is defined as where the target position result and its actual position are not at the same side of the reference nodes. However, this definition is too loose for a sample-based message passing cooperative localisation algorithm. In our definition, it is not a flip ambiguity if the position result meets the following conditions at the same time: (1) the position result and its actual position are at the same side of the reference nodes and (2) the position result must fall within the 95% confidence interval of its prediction distribution obtained by Equation (10). Otherwise, we assume it is a flip ambiguity.

Figure 6 illustrates the ratios of missed alarm and false alarm of the proposed approximate collinear detection criterion with respect to the predefined threshold $\bar{\epsilon}$. In each simulation, 10,000 detections were randomly generated. The ranging noise was $\sigma = 0.5$ m and the predefined threshold $\bar{\epsilon}$ varied from 0.2 m to 2 m. Figure 6(a) shows the ratio of missed alarm to all the flip occurrences. It can be seen from Figure 6(a) that this ratio always maintains a very low probability with different $\bar{\epsilon}$, even in the case of $\bar{\epsilon} = 0.2$ m; the highest ratio is only 0.65%. Figure 6(b) shows the ratio of false alarms to all the non-collinear estimations. It can be observed that as increases, the ratio of false alarms also increases and it is up to 35.1% when $\bar{\epsilon} = 2$ m.

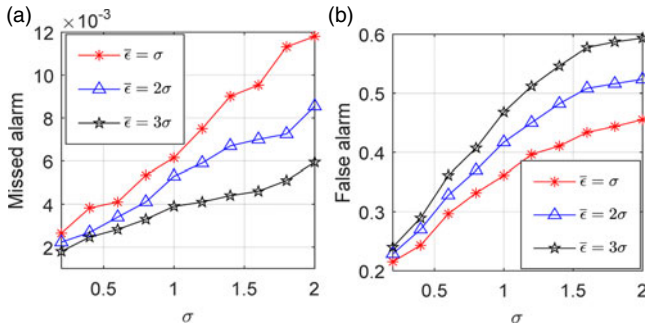


Figure 7. The performance of the approximate collinearity detection criterion, where $\bar{\epsilon} = \sigma/2\sigma/3\sigma$ and σ varies from 0.2 m to 2 m. (a) Missed alarm. (b) False alarm.

The influence of the ranging noise and threshold on the detection performance is shown in Figure 7. The threshold is set to $\bar{\epsilon} = \sigma/2\sigma/3\sigma$, and the ranging noise varies from 0.2 m to 2 m. As is clearly shown in Figure 7, both the ratios of missed alarms and false alarms increase with the increase of σ . When σ is fixed, as $\bar{\epsilon}$ increases, the ratio of missed alarms reduces, but the ratio of false alarms increases. Specifically, taking $\sigma = 2$ m as an example, when the values of $\bar{\epsilon}$ are σ , 2σ and 3σ , the ratios of missed alarms are 1.2%, 0.85% and 0.59%, showing a downtrend, while the ratios of false alarms present an uptrend; they are 45.5%, 52.4% and 59.4%, respectively. The extremely low missed alarms ratio reflects the excellent reliability of the detection scheme. In the case of a large σ , the high false alarms ratio may bring a negative impact on the localizability of the entire network.

5.2. *Small-scale experiment.* To validate the proposed algorithm, a small wireless network with eight nodes was built in the outdoor open area of our university. The UWB node is shown in Figure 8, which consisted of a DWM1000 radio sensor and a 32-bit ARM Cortex-M3 processor as the control unit. The DWM1000 is an IEEE802.15.4-2011 UWB compliant wireless transceiver module based on DecaWave’s DW1000 IC, which supports high-density data transmission within a radius of 20 m. As shown in Figure 8, the anchor node and the static agent node were fixed on the tripod, and the mobile agent node was fixed on the pedestrian’s helmet.

The experimental deployment contained three anchor nodes (Anchors 1, 2, 3), three static agent nodes (Agents 1, 2, 3) and two mobile agent nodes (Agents 4, 5) moving along the trajectories (black line) are shown in Figure 9. In our experiment, the coordinates of the anchors were known exactly, and the agents had no useful information about their location. Throughout the experiment, the connectivity of the five agent nodes are shown in Table 2. It can be seen that the agent node is connected to at most two anchor nodes, and sometimes even three of them are connected to only one anchor, which indicates that none of the agent nodes can estimate its position using the non-cooperative localisation method.

In one of our experiments, the positioning results of the agent nodes are also shown in Figure 9. It can be seen that the position estimates of the three static agent nodes are all close to their true values. In particular, the position estimation of static agent node 1 has the smallest dispersion and the highest accuracy because it can be connected to two anchor nodes and its neighbours have a good geometric distribution. The two blue dotted lines are the estimated trajectories of the mobile agent nodes. It is shown that the estimated

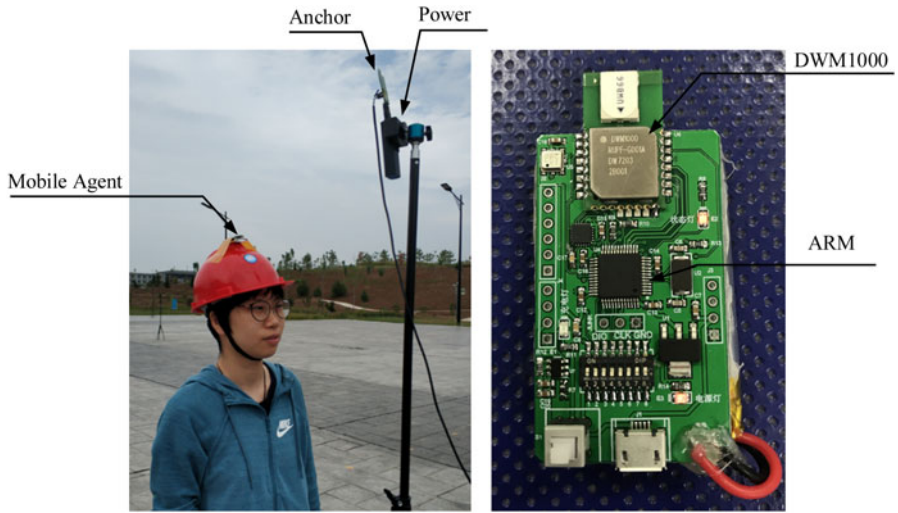


Figure 8. Outdoor positioning environment.

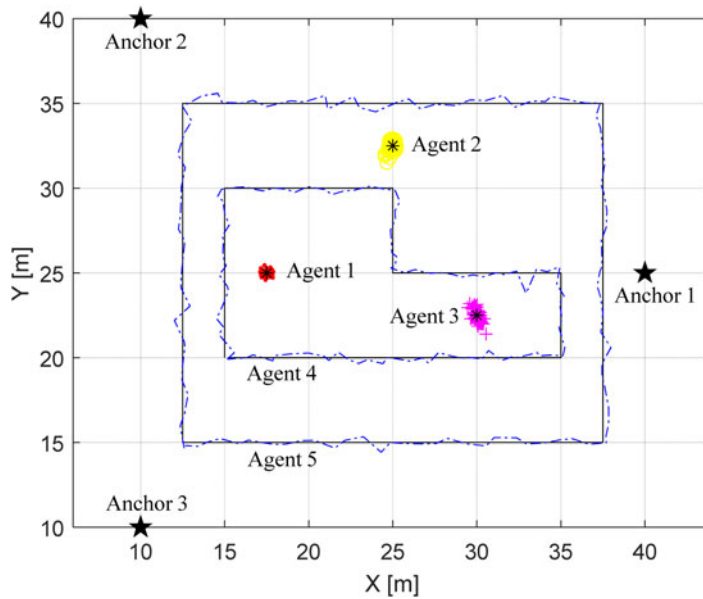


Figure 9. The deployment of the wireless network and the positioning results.

Table 2. Connectivity of the agent nodes in Figure 9.

Agent index	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5
To Anchors (Max)	2	2	1	2	2
To Anchors (Min)	2	2	1	1	1
Total (Max)	6	6	5	6	6
Total (Min)	5	5	4	5	4

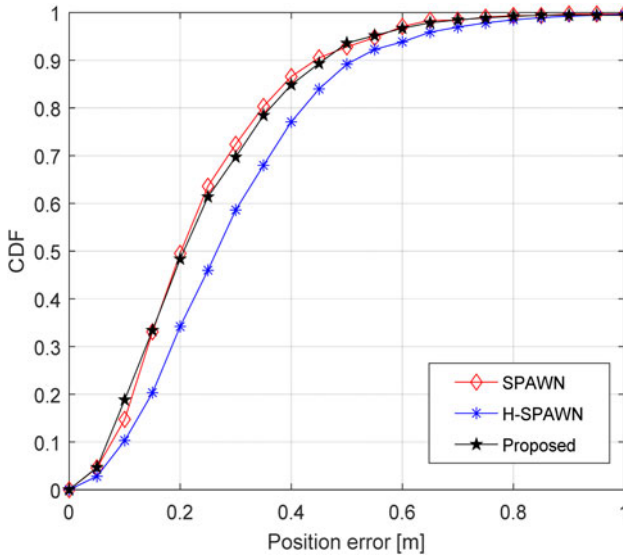


Figure 10. The Cumulative Distribution Function (CDF) of the positioning results of the five agent nodes.

trajectories are consistent with the given paths, indicating that the mobile agents can also achieve accurate position estimation.

The coordinates of the three static agent nodes can be accurately measured manually. Hence, the position error of the static agent at time slot t is defined as $\|\hat{\mathbf{x}}_i^{(t)} - \mathbf{x}\|$. For the mobile agent node, the projection of each estimated position $\hat{\mathbf{x}}_i^{(t)}$ on the given path is first calculated and denoted as $\hat{\mathbf{x}}_{i,proj}^{(t)}$ and then its positioning error is defined as $\|\hat{\mathbf{x}}_i^{(t)} - \hat{\mathbf{x}}_{i,proj}^{(t)}\|$. The CDF of the positioning error corresponding to the five agents is shown in Figure 10. In Figure 10, SPAWN and H-SPAWN are employed as baselines for comparison. As is shown, the performance of the proposed algorithm is very close to that of SPAWN, while it is obviously better than that of H-SPAWN. It can be seen that the ratio of positioning errors less than 0.4 m is 84.85% in the proposed algorithm, which is only 1.71% lower than that achieved by SPAWN but is 7.76% higher than that of H-SPAWN. Note that in this small-scale network, the communication overhead and the complexity of the proposed algorithm is slightly lower than that of H-SPAWN and significantly lower than that of SPAWN. Thus, the results show that our algorithm archives high accuracy with low system overhead in practical applications.

5.3. *Large-scale scenario.* We considered a $100 \times 100 \text{ m}^2$ cooperative localisation area with N anchors and M agents. The maximum sense distance between nodes was $R = 20 \text{ m}$ and the ranging measurement noise was assumed to be an additive white Gaussian noise with zero mean and a standard deviation of σ . The iteration number was assigned as $L = 10$.

A single trial performance of the proposed algorithm is shown in Figures 11 and 12. The simulation configurations were $N = 13$, $M = 100$, $\sigma = 0.5 \text{ m}$ and $\bar{\varepsilon} = \sigma$. The overall performance of the proposed algorithm is illustrated in Figure 11, where ‘ \star ’ and ‘ \cdot ’ denote the actual position of anchor nodes and agent nodes, respectively. The circles illustrate

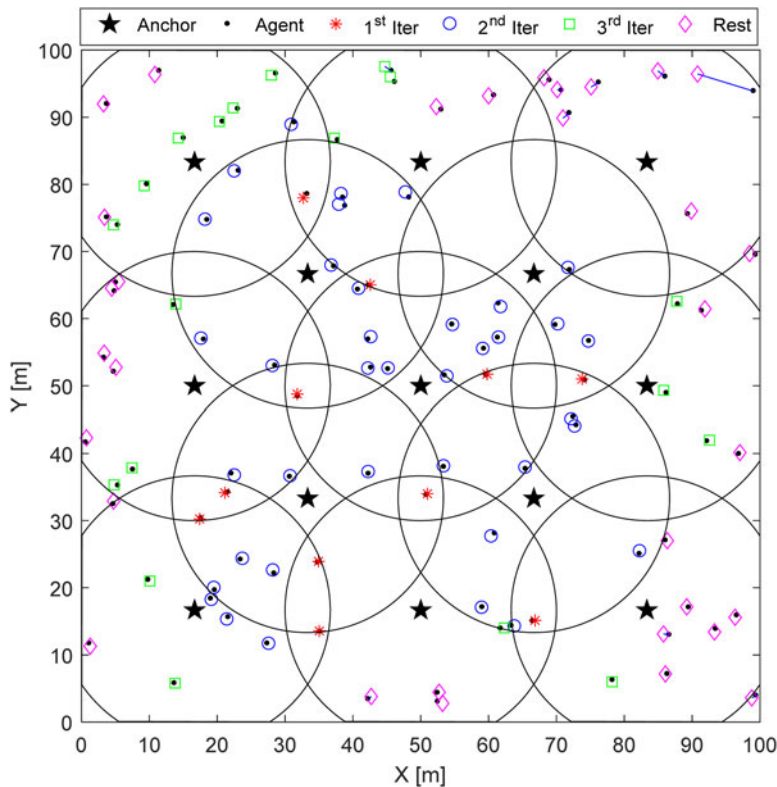


Figure 11. The overall performance of the single trial localisation.

the sense area of the anchor nodes, and the line connects an agent's estimated position and its true position. To reflect the proposed bootstrap percolation scheme intuitively, four notations are used to represent the position results which are located in different iterations. Represented by the red '*', the agents, which are connected to no less than three anchors, can be located in the first iteration with a high accuracy. The blue 'o' represent the agents which are located in the second iteration with the assistance of the red '*'. The green '□' is the position estimation of the agents located in the third iteration. The remaining agents are located in the subsequent iterations gradually and are presented by purple '◇'. As is shown in Figure 11, the agents which are located in the preceding iterations always act as reference nodes to assist the remaining agents in the subsequent iterations. In this way, all the agent nodes will be located layer-by-layer.

A different view is offered in Figure 12, showing the evolution of the positioning error of 100 agents versus the iteration index in the single trial localisation. According to our bootstrap percolation scheme, in each iteration, only the agents satisfying our conditions can be located. As can be seen from Figure 12, as the number of iterations increases, the number of agents that can be located increases in a layer-by-layer manner. In particular, it is observed that in the first iteration, only 11 agents (that is, red '*' agents in Figure 11) can be located and their positioning errors are all less than 1 m. With the assistance of the red '*' agents, another 38 agents (that is, blue 'o' agents in Figure 11) are located in the second

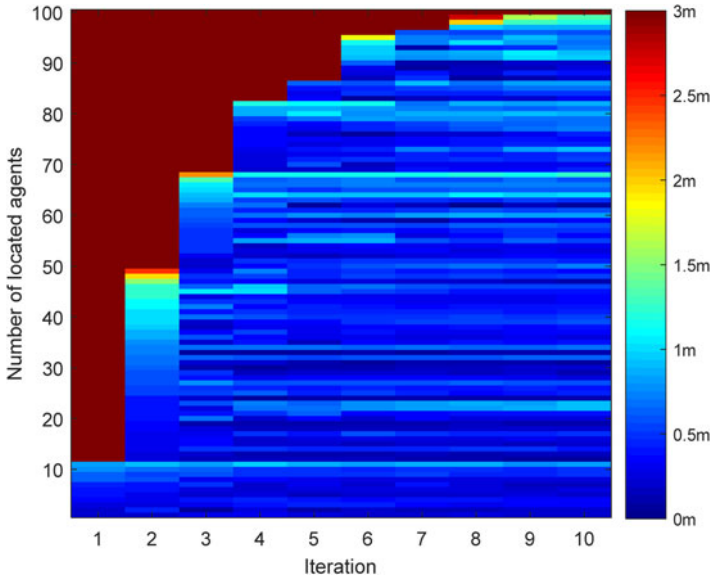


Figure 12. The contour plot of the positioning error of 100 agents versus the iteration index.

iteration. Therefore, in the second iteration, a total of 49 nodes can be located, 46 of which have an accuracy of 1.5 m. This number increased to 67 in the third iteration. Finally, all agents but one, which only has two neighbours, can be located to within 1.4 m by the tenth iteration.

Considering the randomness of both the agents’ deployments and ranging errors, the following results are obtained by averaging 100 Monte Carlo simulations.

With varying ranging noises and detection thresholds, the CDFs and RMSE of the proposed algorithm are shown in Figure 13. Simulation configurations are $N = 13$, $M = 100$, $\sigma = 0.2\text{ m}/0.5\text{ m}/1.0\text{ m}$ and $\bar{\epsilon} = \sigma/2\sigma/3\sigma$. As expected, it can be seen from Figure 13(a) that the influence of the different detection threshold $\bar{\epsilon}$ on the overall positioning performance is almost negligible. The reason is that the ratio of missed alarms remains steady at a very low level with different $\bar{\epsilon}$. Ranging noise determines the positioning accuracy of the proposed algorithm. As is shown, the fraction of positioning errors below 1 m is 98.4% when $\sigma = 0.2\text{ m}$, which is higher than that (95.2%) obtained when $\sigma = 0.5\text{ m}$, as compared to only 71.4% with $\sigma = 1\text{ m}$. From Figure 13(b), we can see that the influence of $\bar{\epsilon}$ on the proposed algorithm is reflected in the iteration process of the positioning. In particular, in the third to fifth iterations, the value of the RMSE increases with the increase of $\bar{\epsilon}$ when σ is constant. This is because the ratio of the false alarms increases with $\bar{\epsilon}$, resulting in a decrease of the number of localised nodes. In the sixth iteration, it has $\mathbb{B}_3^6 = \emptyset$ in most simulations. According to our bootstrap percolation scheme, the set of agent nodes which are located in the sixth iteration becomes $\mathbb{H}^6 = \mathbb{B}_2^6$, reducing the gap of RMSE rapidly. Thus, we can say that our bootstrap percolation scheme is a good supplement to the approximate collinear detection, which can effectively solve the localizability problem caused by the false alarms.

Figure 14 shows the CDF of different algorithms for two networks. Network 1 includes 13 anchors and 100 agents. Network 2 has 13 anchors and 50 agents. The ranging noise is

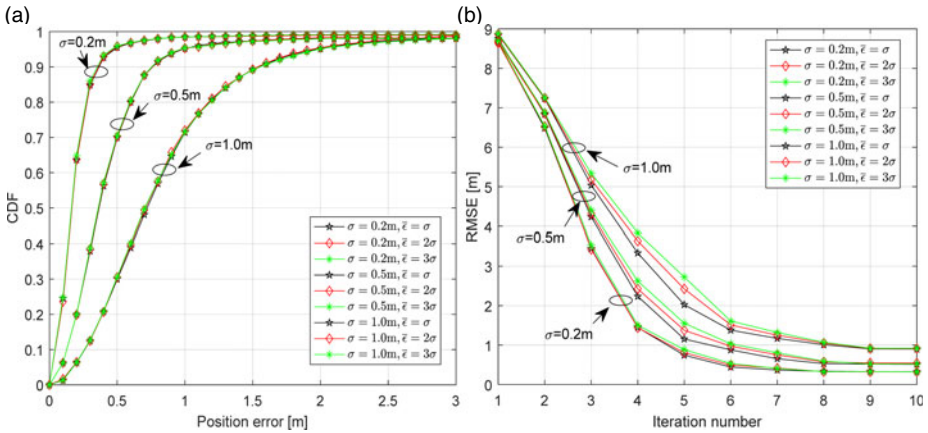


Figure 13. The positioning performance of the proposed algorithm. (a) The CDFs versus the positioning errors; (b) The Root Mean Square Error (RMSE) versus the iteration index.

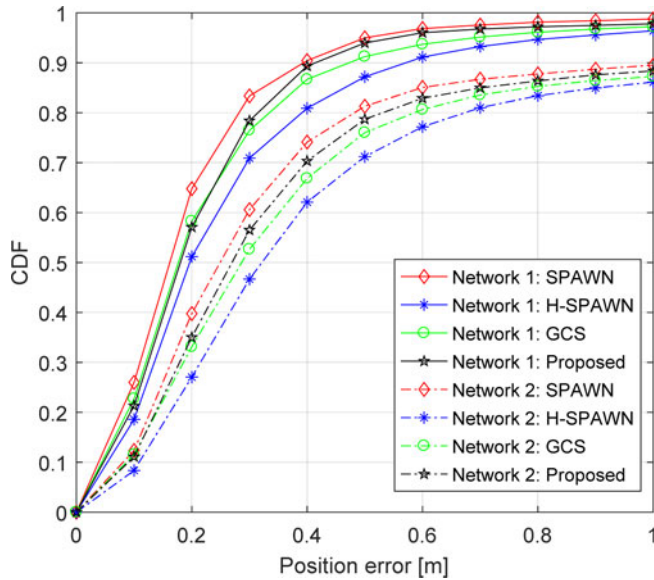


Figure 14. CDF comparison for Network 1 and Network 2.

set to $\sigma = 0.25$ m. In Network 1, on average, the target node is connected to 12.57 neighbour agents, while this number is reduced to 6.28 for Network 2. The performance of Network 1 is better than that of Network 2. It is observed that SPAWN achieves the highest accuracy, which is at the expense of a tremendous system overhead. Both the proposed algorithm and H-SPAWN adopt a Gaussian parametric message passing rule, but the performance of the proposed algorithm is superior to H-SPAWN. The reason is that the bootstrap percolation scheme proposed in this work not only makes full use of all beneficial information, but also efficiently mitigates the error propagation in the entire network. In Network 1, the average Central Processing Unit (CPU) running time required per iteration of the

proposed algorithm, H-SPAWN, GCS and SPAWN is 1.62 s, 5.79 s, 93.57 s and 218.52 s, respectively. Therefore, the results show that compared with H-SPAWN, our algorithm achieves a higher accuracy at roughly a quarter of its execution time.

6. CONCLUSION. In this work, a bootstrap percolation scheme is introduced into a Gaussian parametric message passing-based cooperative localisation algorithm to reduce the system overhead and mitigate the potential error propagation in dense WSNs. In particular, a layer-by-layer message propagation and updating mechanism is designed to locate agents gradually. In each iteration, only the subset with the highest confidence can be located and the others just update their distribution. To solve the potential flip ambiguity problem, we introduce a simple and efficient criterion to identify the approximate collinear geometric configuration of the reference node set. This criterion exploits Mahalanobis distance to evaluate the influence of neighbours' uncertainties on the flip ambiguity. Simulation results demonstrate that this criterion obtains an extremely low missed alarm ratio and our adaptive bootstrap percolation scheme can effectively solve the localizability problem caused by the false alarms ratio. The analysis and simulation results also show that the proposed algorithm reduces the system overhead while keeping the high accuracy of the networks. In our future work, a mixed message passing rule may be exploited to remove the ambiguity of the Gaussian mixture distribution, and the proposed algorithm may be extended to Non-Line Of Sight (NLOS) conditions for a wide range of applications.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. 61174194, 61301094). The authors would like to thank the editor and the anonymous reviewers for their valuable comments which greatly improved the presentation of this paper.

REFERENCES

- Abouzar, P., Michelson, D. G. and Hamdi, M. (2016). RSSI-based Distributed Self-Localization for Wireless Sensor Networks Used in Precision Agriculture. *IEEE Transactions on Wireless Communications*, **15**(10), 6638–6650.
- Bensimhou, M. (2009). N-Dimensional cumulative function, and other useful facts about gaussians and normal densities. https://upload.wikimedia.org/wikipedia/commons/a/a2/Cumulative_function_n_dimensional_Gaussians_12.2013.pdf
- Caceres, M., Penna, F., Wymeersch, H. and Garello, R. (2011). Hybrid Cooperative Positioning Based on Distributed Belief Propagation. *IEEE Journal on Selected Areas in Communications*, **29**(10), 1948–1958.
- Cai, H., Gao, H., Lv, T., Lu, Y. and Su, X. (2014). Propagation Controlled Cooperative Positioning in Wireless Networks Using Bootstrap Percolation. *Proceedings of the 2014 IEEE Global Communications Conference*, Austin, USA.
- Cui, J., Wang, Z., Zhang, C., Zhang, Y. and Zhu, Z. (2017). Message Passing Localization Algorithm Combining BP with VMP for Mobile Wireless Sensor Networks,” *IET Communications*, **11**(7), 1106–1113.
- Dardari, D., Conti, A., Ferner, U., Giorgetti, A. and Win, M. Z. (2009). Ranging with Ultrawide Bandwidth Signals in Multipath Environments. *Proceedings of the IEEE*, **97**(2), 404–426.
- Das, K. and Wymeersch, H. (2012). Censoring for Bayesian Cooperative Positioning in Dense Wireless Networks. *IEEE Journal on Selected Areas in Communications*, **30**(9), 1835–1842.
- De Maesschalck, R., Jouan-Rimbaud, D and Massart, D. L. (2000). The Mahalanobis Distance. *Chemometrics and Intelligent Laboratory Systems*, **50**(1), 1–18.
- Fan, Q., Sun, B., Sun, Y., Wu, Y. and Zhuang, X. (2017). Data Fusion for Indoor Mobile Robot Positioning Based on Tightly Coupled INS/UWB. *The Journal of Navigation*, **70**(5), 1079–1097.

- Georges, H. M., Xiao, Z. and Wang, D. (2016). Hybrid Cooperative Vehicle Positioning Using Distributed Randomized Sigma Point Belief Propagation on Non-Gaussian Noise Distribution. *IEEE Sensors Journal*, **16**(21), 7803–7813.
- Han, S., Yue, J., Meng, W. and Li, C. (2015). A Novel Flip Ambiguities Detection Algorithm for WSN Localization. *Proceedings of the 2015 IEEE Globecom Workshops*, San Diego, USA.
- Ihler, A. T., Fisher, J. W., Moses, R. L. and Willisky, A. S. (2005). Nonparametric Belief Propagation for Self-Localization of Sensor Networks. *IEEE Journal on Selected Areas in Communications*, **23**(4), 809–819.
- Jing, H., Pinchin, J., Hill, C. and Moore, T. (2016). An Adaptive Weighting based on Modified DOP for Collaborative Indoor Positioning. *The Journal of Navigation*, **69**(2), 225–245.
- 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.
- Li, B., Wu, N., Wang, H., Tseng, P. H. and Kuang, J. (2015a). Gaussian Message Passing-Based Cooperative Localization on Factor Graph in Wireless Networks. *Signal Processing*, **111**, 1–12.
- Li, S., Hedley, M. and Collings, I. B. (2015b). New Efficient Indoor Cooperative Localization Algorithm with Empirical Ranging Error Model. *IEEE Journal on Selected Areas in Communications*, **33**(7), 1407–1417.
- Lv, T., Gao, H., Li, X., Yang, S. and Hanzo, L. (2016). Space-Time Hierarchical-Graph Based Cooperative Localization in Wireless Sensor Networks. *IEEE Transactions on Signal Processing*, **64**(2), 322–334.
- Meyer, F., Hlinka, O. and Hlawatsch, F. (2014). Sigma Point Belief Propagation. *IEEE Signal Processing Letter*, **21**(2), 145–149.
- Rantakokko, J., Rydell, J., Stromback, P., Handel, P., Callmer, J. and Tornqvist, D. (2011). Accurate and Reliable Soldier and First Responder Indoor Positioning: Multisensor Systems and Cooperative Localization. *IEEE Wireless Communications*, **18**(2), 10–18.
- Vaze, R. and Gupta, P. (2012). Bounds on Minimum Number of Anchors for Iterative Localization and Its Connections to Bootstrap Percolation. *Proceedings of the 2012 International Conference on Signal Processing and Communications*, Bangalore, India.
- Velde, S. V., Abreu, T. F. and Steendam, H. (2015). Improved Censoring and NLOS Avoidance for Wireless Localization in Dense Networks. *IEEE Journal on Selected Areas in Communications*, **33**(11), 2302–2312.
- Wang, L., Groves, P. and Ziebart, M. (2015). Smartphone Shadow Matching for Better Cross-street GNSS Positioning in Urban Environments. *The Journal of Navigation*, **68**(1), 411–433.
- Wang, X., Liu, Y., Yang, Z., Lu, K. and Luo, J. (2013). OFA: An Optimistic Approach to Conquer Flip Ambiguity in Network Localization. *Computer Networks*, **57**(6), 1529–1544.
- Wymeersch, H., Lien, J. and Win, M. Z. (2009). Cooperative Localization in Wireless Networks. *Proceedings of the IEEE*, **97**(2), 427–450.