Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2007, Article ID 41348, 11 pages
doi:10.1155/2007/41348

Research Article

Positioning Based on Factor Graphs

Christian Mensing and Simon Plass

German Aerospace Center (DLR), Institute of Communications and Navigation, Oberpfaffenhofen, 82234 Wessling, Germany

Received 16 November 2006; Revised 15 March 2007; Accepted 16 April 2007

Recommended by H. Vincent Poor

This paper covers location determination in wireless cellular networks based on time difference of arrival (TDoA) measurements in a factor graphs framework. The resulting nonlinear estimation problem of the localization process for the mobile station cannot be solved analytically. The well-known iterative Gauss-Newton method as standard solution fails to converge for certain geometric constellations and bad initial values, and thus, it is not suitable for a general solution in cellular networks. Therefore, we propose a TDoA positioning algorithm based on factor graphs. Simulation results in terms of root-mean-square errors and cumulative density functions show that this approach achieves very accurate positioning estimates by moderate computational complexity.

1. INTRODUCTION

Positioning in wireless networks became very important in recent years. Services and applications based on accurate knowledge of the location of the mobile station (MS) will play a fundamental role in future wireless systems [1–3]. In addition to vehicle navigation, fraud detection, resource management, automated billing, and further promising applications, it is stated by the United States Federal Communications Commission (FCC) that all wireless service providers have to deliver the location of all enhanced 911 (E911) callers with specified accuracy [4]. Note that a common agreement about location determination of emergency calls in the European Union is not yet well defined and is still in development [5].

MS localization using global navigation satellite systems (GNSSs) such as the global positioning system (GPS) or the future European Galileo system [6, 7] delivers very accurate position information for good environmental conditions. These systems may be a solution for future mass market applications when the problem of high power consumption is resolved and costs are reduced. But nevertheless, the performance loss in indoor areas or urban canyons can be dramatical [8].

Therefore, in this paper we concentrate on determination of the MS location by exploiting the already available communications signals. Generally, this localization process is based on measurements in terms of time of arrival (ToA), time difference of arrival (TDoA), angle of arrival (AoA), and/or received signal strength (RSS) [2], provided by the base stations (BSs) or the MS, where the achievable accuracy is the highest with the timing measurements TDoA or ToA. We will focus on processing TDoA measurements which is a part of the 3GPP standard where it is denoted as observed TDoA (OTDoA) [9]. TDoA is also foreseen for positioning in future fourth-generation (4G) mobile communications systems as it is proposed, for example, within the WINNER project [10, 11].

The localization of the MS leads to a nonlinear estimation problem where no analytical solution is possible [3]. The most popular way to deal with this problem is to use a method based on the iterative Gauss-Newton (GN) algorithm [1, 12]. But this procedure may suffer from convergence problems for certain geometric constellations and inaccurate initial values [13]. To obtain a general solution, we introduce a TDoA positioning method using a factor graphs (FGs) framework in this paper. It provides estimates which achieve high accuracy with low complexity and it is suitable for distributed processing. In [14, 15], Chen et al. proposed a method for solving the positioning problem in an FGs environment using ToA measurements. In [16], they extended their approach to AoA measurements. The still unsolved problem of processing TDoA measurements—with their sophisticated hyperbolic character—will be covered by this paper. Simulation results will be given in terms of root-mean-square errors (RMSEs) and cumulative density

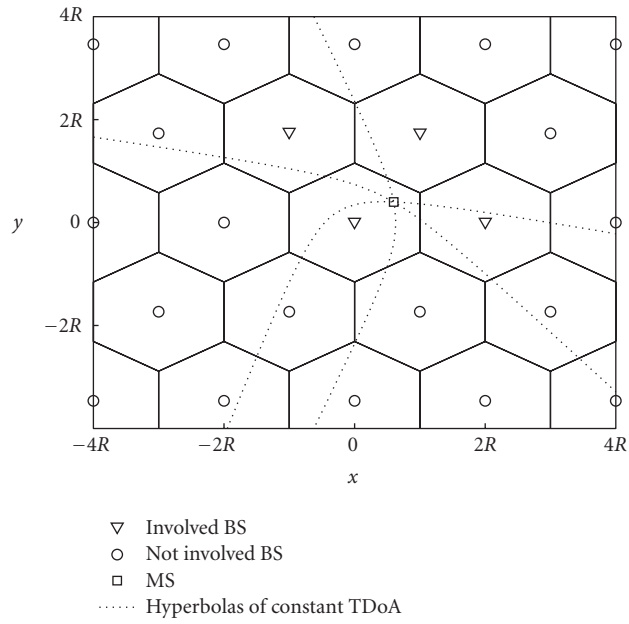

Figure 1: TDoA positioning in cellular networks with $N_{\mathrm{BS}} = 4$ involved BSs.

functions (CDFs). They show the potential of this algorithm in terms of accuracy and computational complexity, outperforming the GN method in cellular networks. Furthermore, a performance comparison with the noniterative Chan-Ho (CH) method [17] is given. Note that the CDFs are an important benchmark in the context of the FCC-E911 requirements. This paper paves the way for a general processing of all kinds of measurements under the joint framework of FGs for future research.

Throughout this paper, vectors and matrices are denoted by lower- and uppercased bold letters. The matrix $\mathbf{I}_n$ is the $n \times n$ identity matrix, the matrix $\mathbf{0}_{n\times m}$ is the $n \times m$ matrix with zeros, the operation "$\otimes$" denotes the Kronecker product, $E\{\cdot\}$ denotes expectation, $(\cdot)^T$ denotes transpose, and $\|\cdot\|_2$ denotes the Euclidean norm.

2. SYSTEM MODEL

The time-synchronized BSs are organized in a cellular network with cell radius R according to Figure 1. For nonsynchronized BSs, the so-called location measurement units (LMUs) are used for processing. The LMUs are associated to the BSs and compensate the missing synchronization of the BS network. The MS is located at $\mathbf{x} = [x, y]^T$ and only the N_{BS} nearest BSs at $\mathbf{x}_\mu$, $\mu \in \{1, 2, \ldots, N_{\mathrm{BS}}\}$ are used for positioning. The distance between the BSs and the MS is given by

$$r_\mu(\mathbf{x}) = \|\mathbf{x}_\mu - \mathbf{x}\|_2 = \sqrt{(x_\mu - x)^2 + (y_\mu - y)^2}. \quad (1)$$

This equation can also be seen as a result of ToA measurements. With ToA, the absolute time for a signal traveling from the BS to the MS or vice versa is measured. It is not even required that all BSs be synchronized with each other, additionally synchronized time knowledge, that is, the time of transmission, is necessary at the MS. In case that no exact time knowledge is available (time offset of the MS), an additional BS is necessary to estimate this offset according to the ToA principle as it is used in GNSSs [6, 7]. Also round-trip delay (RTD) procedures can be chosen to obtain ToAs independent of any synchronization assumptions. But this procedure has the drawback that measurements have to be performed in both uplink and downlink. The propagation time from the BSs to the MS is proportional to the distance. Hence, we get the distance between the MS and all involved BSs. From a geometrical point of view, the MS lies on circles around the BSs. The intersection of these circles gives the position of the terminal.

The problem of processing ToA measurements is the fact that the MS is usually not synchronized to the BSs, and therefore an additional BS is required to estimate the time offset. To avoid this drawback, the TDoAs measure directly the time difference of signals received from various BSs [2, 3], that is, the unknown time offset of the MS with respect to the synchronized BSs is not relevant for TDoA processing. In the geometrical interpretation, the MS lies on hyperbolas with foci at the two related BSs (cf. Figure 1). The intersection gives the position of the MS. Note that TDoAs are defined with respect to an arbitrary chosen reference BS.

In the following, we treat distances and propagation times as equivalent, and thus the TDoAs for BS $\nu \in \{2, 3, \ldots, N_{\mathrm{BS}}\}$ with respect to BS 1 can be written as

$$d_{\nu,1}(\mathbf{x}) = r_\nu(\mathbf{x}) - r_1(\mathbf{x}), \quad (2)$$

where—without loss of generality—we use BS 1 as the reference BS. The $N_{\mathrm{BS}} - 1$ linear independent TDoAs compose the vector

$$\mathbf{d}(\mathbf{x}) = \left[d_{2,1}(\mathbf{x}), d_{3,1}(\mathbf{x}), \ldots, d_{N_{\mathrm{BS}},1}(\mathbf{x})\right]^T, \quad (3)$$

and the corresponding TDoA measurements are given by

$$\mathbf{d} = \left[d_{2,1}, d_{3,1}, \ldots, d_{N_{\mathrm{BS}},1}\right], \quad (4)$$

based on the measurement model

$$\mathbf{d} = \mathbf{d}(\mathbf{x}) + \mathbf{n}, \quad (5)$$

where

$$\mathbf{n} = \left[n_{2,1}, \ldots, n_{N_{\mathrm{BS}},1}\right]^T \quad (6)$$

is zero-mean additive white Gaussian noise (AWGN) [3] with covariance matrix

$$\mathbf{\Sigma_n} = E\{\mathbf{n}\mathbf{n}^{\mathsf{T}}\}. \quad (7)$$

For the solution of the estimation problem for the MS location, it is a common way to follow the weighted nonlinear least-squares approach [3, 12] which minimizes the cost function

$$\varepsilon(\mathbf{x}) = \left(\mathbf{d} - \mathbf{d}(\mathbf{x})\right)^T \mathbf{\Sigma}_\mathbf{n}^{-1} \left(\mathbf{d} - \mathbf{d}(\mathbf{x})\right) \quad (8)$$

with respect to the unknown MS position $\mathbf{x}$ yielding

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}}\, \varepsilon(\mathbf{x}). \tag{9}$$

In the general case, there exists no closed-form solution to the nonlinear two-dimensional optimization problem given by (9), and hence iterative approaches are necessary. A standard approach to deal with (9) is based on the GN algorithm [3, 18]. The GN algorithm linearizes the system model in (5) about some initial value $\mathbf{x}^{(0)}$ yielding

$$\mathbf{d}(\mathbf{x}) \approx \mathbf{d}\left(\mathbf{x}^{(0)}\right) + \boldsymbol{\Phi}(\mathbf{x})\big|_{\mathbf{x}=\mathbf{x}^{(0)}}\left(\mathbf{x} - \mathbf{x}^{(0)}\right), \tag{10}$$

with the elements of the $(N_{\mathrm{BS}} - 1) \times 2$ Jacobian matrix

$$\begin{aligned}\boldsymbol{\Phi}(\mathbf{x}) &= \nabla_{\mathbf{x}}^{T} \otimes \mathbf{d}(\mathbf{x}) \\ &= \begin{bmatrix} \dfrac{x - x_2}{r_2} - \dfrac{x - x_1}{r_1} & \dfrac{y - y_2}{r_2} - \dfrac{y - y_1}{r_1} \\ \dfrac{x - x_3}{r_3} - \dfrac{x - x_1}{r_1} & \dfrac{y - y_3}{r_3} - \dfrac{y - y_1}{r_1} \\ \vdots & \vdots \\ \dfrac{x - x_{N_{\mathrm{BS}}}}{r_{N_{\mathrm{BS}}}} - \dfrac{x - x_1}{r_1} & \dfrac{y - y_{N_{\mathrm{BS}}}}{r_{N_{\mathrm{BS}}}} - \dfrac{y - y_1}{r_1} \end{bmatrix},\end{aligned} \tag{11}$$

where $\nabla_{\mathbf{x}} = [\partial/\partial x, \partial/\partial y]^T$. Afterwards, using (10) and (8), the linear least-squares procedure is applied resulting in the iterated solution

$$\boxed{\begin{aligned}\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \left(\boldsymbol{\Phi}^T\left(\mathbf{x}^{(k)}\right)\boldsymbol{\Sigma}_{\mathbf{n}}^{-1}\boldsymbol{\Phi}\left(\mathbf{x}^{(k)}\right)\right)^{-1} \\ &\quad \cdot \boldsymbol{\Phi}^T\left(\mathbf{x}^{(k)}\right)\boldsymbol{\Sigma}_{\mathbf{n}}^{-1}\left(\mathbf{d} - \mathbf{d}\left(\mathbf{x}^{(k)}\right)\right) \\ &= \mathbf{x}^{(k)} + \mathbf{A}^{(k),-1}\mathbf{g}^{(k)}.\end{aligned}} \tag{12}$$

The GN algorithm provides very fast convergence and accurate estimates for good initial values. For poor initial values and bad geometric conditions the algorithm results in a rank-deficient, and thus noninvertible matrix $\mathbf{A}^{(k)}$ for certain constellations of MS and BSs. In this case, the algorithm diverges [13]. However, a more accurate initial estimate, for example, from a one-step linear least-squares solution as shown in [2], can reduce the divergent behavior of the GN algorithm.

Note that an asymptotically efficient maximum-likelihood (ML) approach to cover this MS positioning problem is not possible in a real-time scenario due to computational complexity. However, we will use the ML solution as reference for the simulation results.

The performance bound for the proposed scenario is given by the Cramer-Rao lower bound (CRLB) [12] for TDoA defined as

$$\boxed{\mathrm{CRLB}(\mathbf{x}) = \mathrm{CRLB}_{\mathrm{TDoA}}(\mathbf{x}) = \sqrt{\mathrm{trace}\left(\boldsymbol{\Phi}^T(\mathbf{x})\boldsymbol{\Sigma}_{\mathbf{n}}^{-1}\boldsymbol{\Phi}(\mathbf{x})\right)^{-1}}} \tag{13}$$

for each MS position where the subscript TDoA is omitted in the following for the sake of simplification. Nevertheless, we are interested in the positioning accuracy for all possible MS locations in the cellular network. Thus, we introduce

$$\overline{\mathrm{CRLB}} = \mathrm{E}_{\mathbf{x}}\{\mathrm{CRLB}(\mathbf{x})\} \tag{14}$$

as mean value of the bound for the whole network.

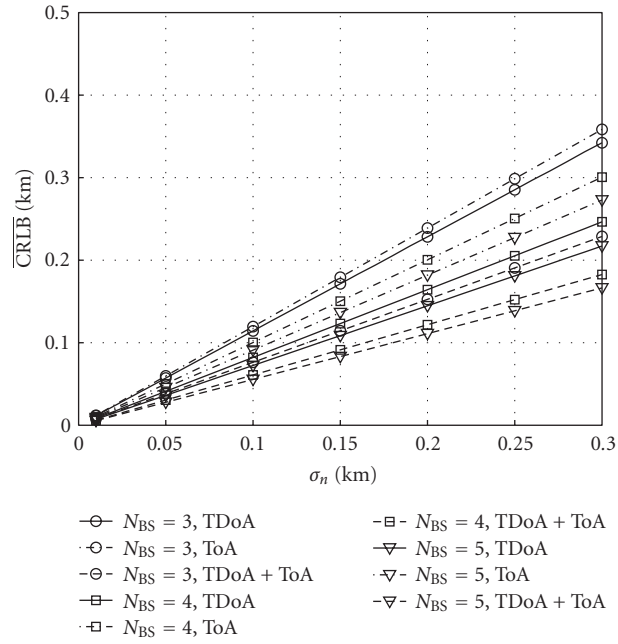

FIGURE 2: $\overline{\mathrm{CRLB}}$ versus σ_n for different positioning methods, $R = 3$ km.

Figure 2 shows the CRLB for TDoA, ToA, and joint TDoA and ToA measurements. The CRLB for ToA is given as

$$\mathrm{CRLB}_{\mathrm{ToA}}(\mathbf{x}) = \sqrt{\mathrm{trace}\left(\boldsymbol{\Psi}^T(\mathbf{x})\boldsymbol{\Sigma}_{\mathbf{n},\mathrm{ToA}}^{-1}\boldsymbol{\Psi}(\mathbf{x})\right)^{-1}}, \tag{15}$$

with

$$\boldsymbol{\Psi}(\mathbf{x}) = \nabla_{\mathbf{x}}^{T} \otimes \mathbf{r}(\mathbf{x}) = \begin{bmatrix} \dfrac{x - x_1}{r_1} & \dfrac{y - y_1}{r_1} \\ \dfrac{x - x_2}{r_2} & \dfrac{y - y_2}{r_2} \\ \vdots & \vdots \\ \dfrac{x - x_{N_{\mathrm{BS}}}}{r_{N_{\mathrm{BS}}}} & \dfrac{y - y_{N_{\mathrm{BS}}}}{r_{N_{\mathrm{BS}}}} \end{bmatrix}, \tag{16}$$

where

$$\mathbf{r}(\mathbf{x}) = \left[r_1(\mathbf{x}), r_2(\mathbf{x}), \ldots, r_{N_{\mathrm{BS}}}(\mathbf{x})\right]^T, \tag{17}$$

and $\boldsymbol{\Sigma}_{\mathbf{n},\mathrm{ToA}}$ is the covariance matrix of the noise for ToA measurements. Equivalently, the CRLB for joint TDoA and ToA measurements can be calculated as

$$\begin{aligned}&\mathrm{CRLB}_{\mathrm{TDoA + ToA}}(\mathbf{x}) \\ &= \sqrt{\mathrm{trace}\left(\begin{bmatrix} \boldsymbol{\Phi}\ (\mathbf{x}) \\ \boldsymbol{\Psi}\ (\mathbf{x}) \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{n}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\mathbf{n},\mathrm{ToA}} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Phi}\ (\mathbf{x}) \\ \boldsymbol{\Psi}\ (\mathbf{x}) \end{bmatrix}\right)^{-1}}.\end{aligned} \tag{18}$$

For the simulations, we assume that the noise variance is the same for each measurement, that is, we use $\mathbf{\Sigma_n} = \sigma_n^2 \mathbf{I}_{N_{\mathrm{BS}}-1}$ (perfect power control). It should be pointed out that for ToA and also joint TDoA + ToA procedures (Figure 2), the CRLB can be achieved with simple methods, for example, based on the GN algorithm. However, for TDoA with the hyperbolic character of the measurements, the effort to the algorithms is much higher to achieve CRLB over the whole network and more sophisticated methods—as in the following proposed FG-based approach—are necessary.

Note that all considered algorithms in this paper are not restricted to any assumptions about the source of the measurements. They work for both uplink and downlink measurements and are independent of the underlying wireless cellular network.

3. POSITIONING BASED ON FACTOR GRAPHS

Historically, FGs as a generalization of Tanner graphs come from coding theory and were used for decoding of low-density parity check (LDPC) or concatenated (turbo) codes. But additionally, there exist a lot of algorithms which can be described in an FGs framework [19], for example, Kalman filters or Fourier transforms. In [14, 15], Chen et al. proposed a method for solving the positioning problem in an FGs environment using ToA measurements. In [16], they extended their method to AoA measurements. In this section, we present the solution for FG-based positioning using TDoA measurements with—compared to ToA and AoA—their more complicated hyperbolic character. In the following, we give a short overview of basic principles regarding FGs theory. Afterwards, we describe necessary geometric fundamentals for the proposed procedure. Finally, the TDoA positioning algorithm using FGs is derived in detail.

3.1. Factor graphs and the sum-product algorithm

An FG is a bipartite graph that in its original sense can describe the structure of a factorization [19]. If we assume as an example the function $f(x_1, x_2, x_3, x_4)$ which can be factorized in

$$f(x_1, x_2, x_3, x_4) = f_1(x_1) f_2(x_1, x_2, x_3) f_3(x_3, x_4), \tag{19}$$

the structure of this factorization can be expressed by an FG. The bipartite FG consists of variable nodes for each variable x_ν occurring in the function, factor nodes for each local function f_μ of f, and edges connecting variable nodes x_ν with factor nodes f_μ, if and only if x_ν is a function of f_μ [19]. The corresponding FG for the example (cf. (19)) can be seen in Figure 3.

Often, we are interested in a marginalization of such a structured function, that is, to find a function only depending on one of the unknowns describing for example the extrinsic soft information in a decoding framework. A systematic way to do so is the application of the so-called sum-product algorithm (SPA). It is a message passing algorithm working on the FG. There is no general assumption about these messages, for example, they can take the form of soft

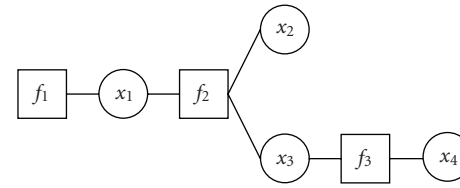

Figure 3: Example for a factor graph.

information or probability density functions (PDFs). For example, in a decoding scenario of Hamming codes, the factor nodes describe the parity check constraints and the messages passed in the FG consist of the extrinsic soft information.

In the following, we describe the fundamental rules of the SPA and refer to [19] for more details. Generally, we differentiate between messages passed from variable nodes to factor nodes and vice versa. According to the SPA, a message passing from a variable node x to a factor node f should be calculated as

$$L_{x\to f}(x) = \prod_{h\in n(x)\setminus\{f\}} L_{h\to x}(x), \tag{20}$$

where $n(x)$ denotes the set of neighbors of a given node x in the FG. Note that the product in (20) should be interpreted in a more abstract way, depending on type and structure of the messages. The rule for computing a message from local node f to variable node x is given as

$$L_{f\to x}(x) = \sum_{\sim\{x\}} \left(f(X) \prod_{y\in n(f)\setminus\{x\}} L_{y\to f}(y) \right), \tag{21}$$

with the set of arguments of the function f defined as $X = n(f)$, and the so-called summary operator $\sum_{\sim\{x\}}$ indicating the variables being not summed over. With these two rules—after an initialization step for the nodes at the edges—all messages in the FG can be calculated step by step. For the final calculation of the marginalization of the variables, we use the termination rule

$$L(x) = \prod_{h\in n(x)} L_{h\to x}(x), \tag{22}$$

that is, the multiplication of all incident messages to this variable node, depending only on one variable.

Generally, it is differentiated between FGs with and without cycles. For cycle-free FGs, the optimum performance—depending on the quality criterion—can be achieved, but for FGs with cycles only an approximation of the optimum solution is obtained. Besides, in case of FGs with cycles usually adequate scheduling algorithms are necessary to determine the messages in a specific order. Nevertheless, there often exists no optimum classical solution or it is associated with too high computational complexity (cf. LDPC or turbo codes). Therefore, a solution based on an FG with cycles may be the only reasonable way to deal with the problem.

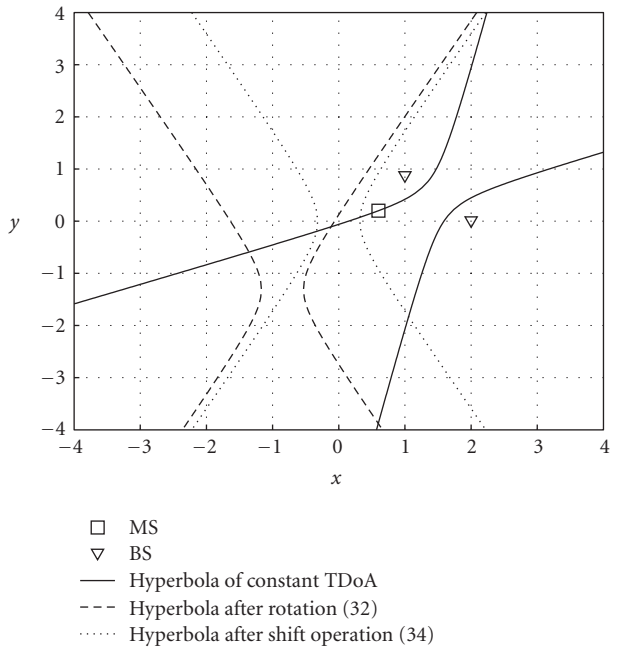

Figure 4: Hyperbola processing.

3.2. Geometric fundamentals

To derive the algorithm of TDoA-based positioning using FGs, we need some fundamental calculus taken from theory of conic sections and shown in the following. The aim of this procedure is to provide the local constraints of the factor nodes in the FG. The idea is to process x and y coordinates mostly independent. Information is only exchanged at the so-called mapping factor nodes. From a geometric point of view, the proceeding is based on a principal axis transformation by rotation, a shift operation, and finally the mapping operation where the original hyperbola equation is transformed in a suitable way for FG processing.

As first step, each TDoA equation (2)—under assumption of the measurement model defined in (5)—can be rewritten as

$$\begin{aligned} &a_{\nu,11}x^2 + a_{\nu,22}y^2 + (a_{\nu,12} + a_{\nu,21})xy \\ &\quad + (a_{\nu,01} + a_{\nu,10})x + (a_{\nu,02} + a_{\nu,20})y \\ &\quad + a_{\nu,00} = 0 \end{aligned} \tag{23}$$

for all $N_{\mathrm{BS}} - 1$ TDoAs, using simple algebraic operations. Note that due to squaring operations a second hyperbola branch—compared to the original TDoA equation (2)—appears (cf. Figure 4), where the MS does not lie on. Nevertheless, this ambiguity can be resolved by observing the signs of the corresponding TDoAs and does not restrict the performance of the later derived algorithm. The coefficients $a_{\nu,ij}$, $i, j \in \{0, 1, 2\}$, in (23) can simply be computed in dependence on the known BS positions and measurements. The quadratic equation (23) can be written in matrix-vector notation resulting in

$$\mathbf{x}^T\mathbf{A}_\nu\mathbf{x} + \mathbf{a}_\nu^T\mathbf{x} + a_{\nu,00} = 0 \tag{24}$$

for all $\nu \in \{2, 3, \ldots, N_{\mathrm{BS}}\}$ TDoAs related to reference BS 1, using the quadratic form $\mathbf{x}^T\mathbf{A}_\nu\mathbf{x}$ with

$$\mathbf{A}_\nu = \begin{bmatrix} a_{\nu,11} & a_{\nu,12} \\ a_{\nu,21} & a_{\nu,22} \end{bmatrix} \tag{25}$$

and the vector

$$\mathbf{a}_\nu = \left[a_{\nu,10}, a_{\nu,20}\right]^T. \tag{26}$$

Equation (24) can be diagonalized by application of an eigenvalue decomposition of the matrix

$$\mathbf{A}_\nu = \mathbf{U}_\nu\mathbf{\Lambda}_\nu\mathbf{U}_\nu^T, \tag{27}$$

where

$$\mathbf{\Lambda}_\nu = \begin{bmatrix} \lambda_{\nu,1} & 0 \\ 0 & \lambda_{\nu,2} \end{bmatrix} \tag{28}$$

is the diagonal matrix composed of the eigenvalues, and

$$\mathbf{U}_\nu = \begin{bmatrix} \mathbf{u}_{\nu,1} & \mathbf{u}_{\nu,2} \end{bmatrix} \tag{29}$$

is a unitary matrix with the corresponding eigenvectors. For later purposes, we choose the order of the eigenvalues in such a way that

$$\operatorname{sign}(\lambda_{\nu,1}) \operatorname{sign}(B_\nu) = 1 \tag{30}$$

is fulfilled, where

$$B_\nu = \mathbf{a}_\nu^T\mathbf{U}_\nu\mathbf{\Lambda}_\nu^{-1}\mathbf{U}_\nu^T\mathbf{a}_\nu - a_{\nu,00}. \tag{31}$$

With the substitution

$$\boxed{\mathbf{x} = \mathbf{U}_\nu\mathbf{x}'_\nu} \tag{32}$$

in (24)—describing the rotation—we obtain

$$\left(\mathbf{x}'_\nu\right)^T\mathbf{\Lambda}_\nu\mathbf{x}'_\nu + \mathbf{a}_\nu^T\mathbf{U}_\nu\mathbf{x}'_\nu + a_{00} = 0, \tag{33}$$

that is, a diagonalized system which can be seen as the original system in a new coordinate plan (cf. Figure 4). In a second step, the rotated hyperbola is shifted around the origin. For this purpose, we have to differentiate between two cases depending on the character of the eigenvalues. In the first case ($\lambda_{\nu,1} \neq 0$, $\lambda_{\nu,2} \neq 0$), we can do the second substitution (shift operation)

$$\boxed{\mathbf{x}'_\nu = \mathbf{x}''_\nu - \frac{1}{2}\mathbf{\Lambda}_\nu^{-1}\mathbf{U}_\nu^T\mathbf{a}_\nu} \tag{34}$$

in (33), yielding the hyperbola equation in main position (cf. Figure 4) given as

$$\left(\mathbf{x}''_\nu\right)^T\mathbf{B}_\nu\mathbf{x}''_\nu - 1 = 0, \tag{35}$$

with

$$\mathbf{B}_\nu = \begin{bmatrix} \dfrac{B_\nu}{\lambda_{\nu,1}} & 0 \\ 0 & \dfrac{B_\nu}{\lambda_{\nu,2}} \end{bmatrix}. \tag{36}$$

Note that

$$\operatorname{sign}(\mathbf{B}_\nu) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{37}$$

is always valid by construction (30) which confirms the hyperbolic character. In case that one eigenvalue is equal to zero (by construction: $\lambda_{\nu,1} \neq 0$, $\lambda_{\nu,2} = 0$), the two hyperbola branches degenerate into a line, yielding the other case of the second substitution

$$\mathbf{x}'_\nu = \begin{bmatrix} x''_\nu - \dfrac{\mathbf{a}_\nu^T \mathbf{u}_{\nu,1}}{2\lambda_{\nu,1}} \\ y''_\nu \end{bmatrix}, \tag{38}$$

and the degenerated hyperbola equation becomes

$$(x''_\nu)^2 = 0. \tag{39}$$

This case occurs if the corresponding TDoA is equal to zero. Note that the case that both eigenvalues are equal to zero is not relevant for realistic BSs and MS constellations.

As last step, we define a mapping operation which will be used in the mapping factor nodes of the FG to exchange information in x and y directions. According to (35), we define

$$\boxed{\begin{aligned} x''_\nu &= \pm\sqrt{\frac{\lambda_{\nu,1}}{B_\nu} - \frac{\lambda_{\nu,1}}{\lambda_{\nu,2}}(y''_\nu)^2}, \\ y''_\nu &= \pm\sqrt{\frac{\lambda_{\nu,2}}{B_\nu} - \frac{\lambda_{\nu,2}}{\lambda_{\nu,1}}(x''_\nu)^2}. \end{aligned}} \tag{40}$$

3.3. TDoA positioning algorithm based on factor graphs

The aim of this positioning algorithm using FGs is to determine the location of the MS by processing the measured TDoAs with their statistic properties and the known BS positions. It breaks down the general high-complex problem in several low-complex subproblems that can be solved in a distributed way and finds the solution—starting with an initial guess—iteratively. The corresponding FG is depicted in Figure 5. The factor nodes are given by the substitution and mapping operations defined in the previous section (see (32), (33), and (34)) and can be seen as constraint nodes among the variables. The rotation (R) and shift (S) operations process the messages of x and y coordinates independently. In the mapping (M) nodes, information between x and y coordinates is exchanged. The messages that are passed around the FG are defined as PDFs for the corresponding variables in the variable nodes. In our investigations, we assume Gaussian distributions for these PDFs according to

$$\mathcal{N}(x, m, \sigma^2) \sim \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right), \tag{41}$$

for a random variable x with mean value m and variance σ^2. This assumption simplifies the calculations performed by the

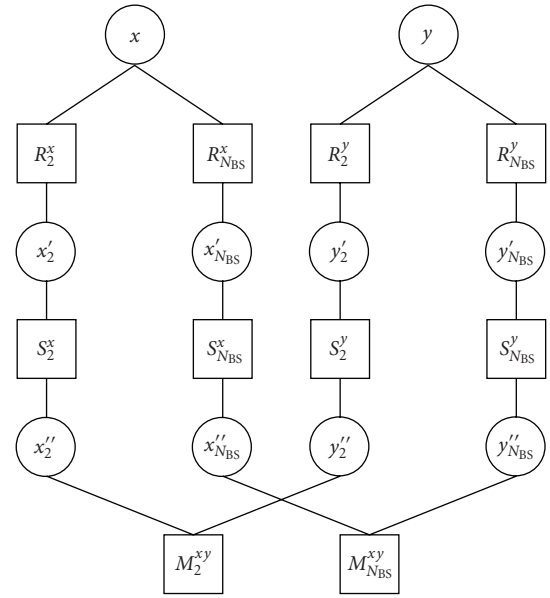

Figure 5: Factor graph for TDoA positioning.

SPA considerably. After the initialization of the FG in a suitable way [15], the rules of SPA can be applied straightforwardly. Furthermore, in our derivation we need two general rules [19] for random variables with Gaussian distributions. At first, the relation

$$\prod_{n=1}^{N} \mathcal{N}(x, m_n, \sigma_n^2) \sim \mathcal{N}(x, m_{\mathrm{P}}, \sigma_{\mathrm{P}}^2) \tag{42}$$

holds, where the mean value of the resulting Gaussian distribution can be calculated as

$$m_{\mathrm{P}} = \sigma_{\mathrm{P}}^2 \sum_{n=1}^{N} \frac{m_n}{\sigma_n^2}, \tag{43}$$

and the variance is given as

$$\sigma_{\mathrm{P}}^2 = \frac{1}{\sum_{n=1}^{N} 1/\sigma_n^2}. \tag{44}$$

Secondly, we will make use of the integration rule

$$\begin{aligned} &\int_{x=-\infty}^{\infty} \mathcal{N}(x, m_1, \sigma_1^2)\, \mathcal{N}(y, \alpha x + m_2, \sigma_2^2)\, dx \\ &\quad \sim \mathcal{N}(y, \alpha m_1 + m_2, \alpha^2\sigma_1^2 + \sigma_2^2). \end{aligned} \tag{45}$$

In the following, we show the necessary message passing operations. Note that the iteration index is omitted here for the sake of simplification and that the summary operator (cf. (21)) is replaced by an integration operator due to the Gaussian character of the messages. We start at the messages passed from the variable node x to factor nodes R_ν^x. According to SPA, we obtain

$$L_{x \to R_\nu^x}(x) \sim \prod_{\mu \neq \nu} L_{R_\mu^x \to x}(x), \tag{46}$$

that is, the multiplication of all incident messages which can be calculated using (42). For the messages from the factor nodes R_ν^x to the variable nodes x'_ν, information about R_ν^x is essential. This is given by the rotation (R) operation as first substitution (32), and under assumption of Gaussian distributions, we obtain

$$f_{R_\nu^x}(x, x'_\nu) \sim \mathcal{N}\left(x'_\nu, \mathbf{u}_{\nu,1}^T \mathbf{x}, \sigma_{x'_\nu}^2\right) \tag{47}$$

as constraint rule for the factor node where the variance is set to the variance in x-direction of the noisy observations, that is, $\sigma_{x'_\nu}^2 = \sigma_x^2$ which can be derived from (7) [15]. The SPA yields

$$L_{R_\nu^x \to x'_\nu}(x'_\nu) \sim \int_{x=-\infty}^{\infty} f_{R_\nu^x}(x, x'_\nu) L_{x \to R_\nu^x}(x) dx, \tag{48}$$

which can be computed with (45). We proceed by calculating the messages to the shift operation (S) factor node, simply given as

$$L_{x'_\nu \to S_\nu^x}(x'_\nu) = L_{R_\nu^x \to x'_\nu}(x'_\nu). \tag{49}$$

The messages from the shift operation nodes S_ν^x with the constraint (cf. (34))

$$f_{S_\nu^x}(x'_\nu, x''_\nu) \sim \mathcal{N}\left(x''_\nu, x'_\nu + \frac{1}{2\lambda_{\nu,1}} \mathbf{u}_{\nu,1}^T \mathbf{a}_\nu, \sigma_{x''_\nu}^2\right) \tag{50}$$

to the variable nodes x''_ν ($\sigma_{x''_\nu}^2 = \sigma_x^2$) are obtained by

$$L_{S_\nu^x \to x''_\nu}(x''_\nu) \sim \int_{x'_\nu=-\infty}^{\infty} f_{S_\nu^x}(x'_\nu, x''_\nu) L_{x'_\nu \to S_\nu^x}(x'_\nu) dx'_\nu \tag{51}$$

using (45). The messages to the mapping (M) nodes M_ν^{xy} can simply be calculated as

$$L_{x''_\nu \to M_\nu^{xy}}(x''_\nu) = L_{S_\nu^x \to x''_\nu}(x''_\nu). \tag{52}$$

A very important node is the mapping node, where information between x and y coordinates is exchanged. It is based on the mapping operations defined in (40), but to fulfil the Gaussian assumption in the corresponding factor node, a Gaussian approximation similar as that shown in [15] has to be performed. Additionally, several cases due to sign ambiguities have to be distinguished. Thus, in this paper we give only the general formula according to SPA obtained by

$$L_{M_\nu^{xy} \to y''_\nu}(y''_\nu) \sim \int_{x''_\nu=-\infty}^{\infty} f_{M_\nu^{xy}}(x''_\nu, y''_\nu) L_{x''_\nu \to M_\nu^{xy}}(x''_\nu) dx''_\nu, \tag{53}$$

where $f_{M_\nu^{xy}}(x''_\nu, y''_\nu)$ describes the mapping constraint (40). The necessary message calculations from the mapping nodes back to the variable nodes x and y and the processing in the y branch of Figure 5 correspond to the steps described above. After initialization, the messages are calculated iteratively up to convergence. We emphasize that due to the Gaussian character of the messages and the possibility of distributed computing, the processing effort is limited to simple operations.

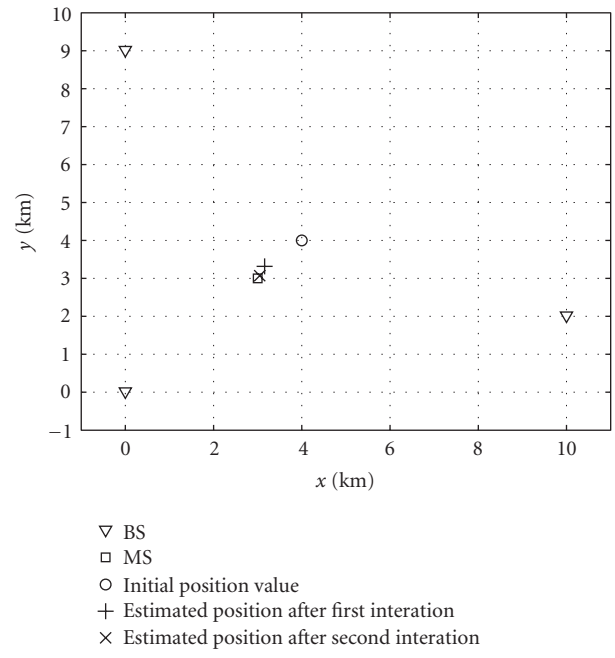

FIGURE 6: Example for the FG positioning algorithm with $N_{\text{BS}} = 3$ BSs.

In a final termination step, the marginalization of the coordinate variable nodes x and y can easily be computed by

$$L(x) = \prod_\nu L_{R_\nu^x \to x}(x) \tag{54}$$

for the x coordinate. The final estimates $\hat{x} = x^{(K)}$ after K iteration steps are given by the mean values of the Gaussian distributions according to (54). Note that also the variances in x and y directions are provided by the algorithm.

We are aware of the fact that the here proposed FG has cycles, and therefore the estimates are only an approximation of the optimum solution. But simulation results show the near-optimum behavior of the algorithm for the general scenario considered in this paper. Analytical investigations on the convergence behavior of this FG with cycles are hard to establish because the occurring cycles are very short.

3.4. Implementation example

To demonstrate the functionality of the FG-based positioning algorithm, we show a simple example with $N_{\text{BS}} = 3$ involved BSs at $\mathbf{x}_1 = [0, 0]^T$, $\mathbf{x}_2 = [0, 9]^T$, and $\mathbf{x}_3 = [10, 2]^T$ (cf. Figure 6). Hence, two TDoA measurements for the MS at $\mathbf{x} = [3, 3]^T$ are available that can be calculated as $\mathbf{d}(\mathbf{x}) = [2.47, 2.83]^T$. The noise vector for these measurements is given as $\mathbf{n} = [0.2, -0.2]^T$ with variances in x- and y-components as $\sigma_x^2 = \sigma_y^2 = 0.1$, and the initial value is at $\mathbf{x}^{(0)} = [4, 4]^T$.

In the following, we describe the required calculations in more detail for this example. For processing the

x-components of the first TDoA measurement, we need the matrix and vector

$$\mathbf{A}_1 = \begin{bmatrix} 7.11 & 0 \\ 0 & -73.89 \end{bmatrix}, \qquad \mathbf{a}_1 = [0, 665.05]^T. \tag{55}$$

The resulting eigenvalues are $\lambda_{1,1} = -73.89$ and $\lambda_{1,2} = 7.11$, and the eigenvector for the x-component is $\mathbf{u}_{1,1} = [0, 1]^T$. Finally, the scalar value $B_1 = -131.26$ is required.

With this information, we can start the algorithm in the previous section. It is initialized with

$$L_{x\to R_2^x}(x) \sim \mathcal{N}(x, 4, 0.1), \tag{56}$$

that is, for the x-component the message from the variable node x to the factor node R_2^x includes in the first step a Gaussian distribution with mean value as the x-component of the initial value $\mathbf{x}^{(0)}$, and an assumed variance of $\sigma_x^2 = 0.1$. With knowledge of the eigenvectors, the constraint rule for the rotation factor node is given as

$$f_{R_2^x}(x, x_2') \sim \mathcal{N}(x_2', 4, 0.1). \tag{57}$$

Hence, the merged Gaussian distributions for the message from the rotation factor node to the variable node x_2' can be calculated as

$$L_{R_2^x\to x_2'}(x_2') \sim \mathcal{N}(x_2', 8, 0.2), \tag{58}$$

where we have used the relation in (45) with $\alpha = 1$. The message to the shift factor node is simply given as

$$L_{x_2'\to S_2^x}(x_2') = L_{R_2^x\to x_2'}(x_2') \sim \mathcal{N}(x_2', 8, 0.2). \tag{59}$$

Similar to the rotation rule, the shift rule can be calculated as

$$f_{S_2^x}(x_2', x_2'') \sim \mathcal{N}(x_2'', -6.5, 0.1), \tag{60}$$

which is further used to calculate the message to the variable node x_2''. We obtain

$$L_{S_2^x\to x_2''}(x_2'') \sim \mathcal{N}(x_2'', 1.5, 0.3), \tag{61}$$

again using (45). The messages to the mapping factor node are simply

$$L_{x_2''\to M_2^{xy}}(x_2'') = L_{S_2^x\to x_2''}(x_2'') \sim \mathcal{N}(x_2'', 1.5, 0.3). \tag{62}$$

Using (40), the mean value for the mapping node can be calculated as 3.36. The corresponding variance is given as 0.17. Hence, we obtain

$$L_{M_2^{xy}\to y_2''}(y_2'') \sim \mathcal{N}(y_2'', 3.36, 0.17). \tag{63}$$

On the backward step from y_2'' to y, the described operations are very similar, and we end up at

$$L_{R_2^y\to y}(y) \sim \mathcal{N}(y, 3.16, 0.36). \tag{64}$$

Calculating the values from y to x for the first TDoA measurement, we obtain

$$L_{R_2^x\to x}(x) \sim \mathcal{N}(x, 3.86, 0.41). \tag{65}$$

The values for the second TDoA can be computed as

$$\begin{aligned} L_{R_3^y\to y}(y) &\sim \mathcal{N}(y, 3.36, 0.32), \\ L_{R_3^x\to x}(x) &\sim \mathcal{N}(x, 2.58, 0.31). \end{aligned} \tag{66}$$

With these messages, mean value and variance of the estimated position can be calculated using relation (42). This yield the improved estimation after the first iteration

$$\mathbf{x}^{(1)} \sim \mathcal{N}\left(\mathbf{x}, \begin{bmatrix} 3.19 \\ 3.28 \end{bmatrix}, \begin{bmatrix} 0.18 \\ 0.17 \end{bmatrix}\right). \tag{67}$$

Of course, more iterations can be performed to further improve the performance (cf. Figure 6).

4. SIMULATION RESULTS

We test the proposed algorithms in a cellular network with cell radius $R = 3$ km and assume constant noise power for all involved links from the BSs to the MS, that is, $\mathbf{\Sigma}_n = \sigma_n^2 \mathbf{I}_{N_{\mathrm{BS}}-1}$.

Figures 7–9 show CRLB($\mathbf{x}$) (cf. (13)) using $N_{\mathrm{BS}} = \{3, 4, 5\}$ for positioning. We observe that, for example, for $N_{\mathrm{BS}} = 3$ near the BSs and on the links between the BSs the positioning performance is restricted due to geometric constellation. In these cases, we can expect limited performance of the algorithms. We further can see that the performance increases when more BSs are involved in the localization process.

In Figure 10, the performance of the investigated algorithms is analyzed for $N_{\mathrm{BS}} = 3$ and $\sigma_n = 0.2$ km. Initial value for the iterative algorithms is the mean value of the positions of all involved BSs, that is,

$$\mathbf{x}^{(0)} = \frac{1}{N_{\mathrm{BS}}} \sum_{\nu=1}^{N_{\mathrm{BS}}} \mathbf{x}_\nu. \tag{68}$$

We compare $\overline{\mathrm{CRLB}}$ (cf. (14)) with the achievable RMSE for the algorithms defined as

$$\mathrm{RMSE} = \sqrt{\mathrm{E}_{\mathbf{x}}\{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2\}} \geq \overline{\mathrm{CRLB}} \tag{69}$$

and averaged over several MS positions and noise realizations, where $\hat{\mathbf{x}} = \mathbf{x}^{(K)}$ is the estimate provided by the iterative algorithms after K iteration steps. The GN algorithm provides very fast convergence and accurate estimates for good initial values. For poor initial values and bad geometric conditions (e.g., at the cell edge or near the BSs), the algorithm diverges [13]. Therefore, in these cases, the resulting estimate is set to $\hat{\mathbf{x}} = \mathbf{x}^{(0)}$ to show the loss with respect to $\overline{\mathrm{CRLB}}$. Anyway, for perfect conditions, GN provides fast convergence. The FG algorithm converges after $K_{\mathrm{FG}} = 8$ iteration steps and reaches nearly $\overline{\mathrm{CRLB}}$. Additionally, in Figure 10 the needed floating point operations (FLOPs) as measure for computational complexity are depicted for both algorithms. Obviously, FG offers a better performance compared to GN by just slightly increased complexity.

Figure 11 shows a performance comparison of the FG algorithm for various numbers of involved BSs. It can be

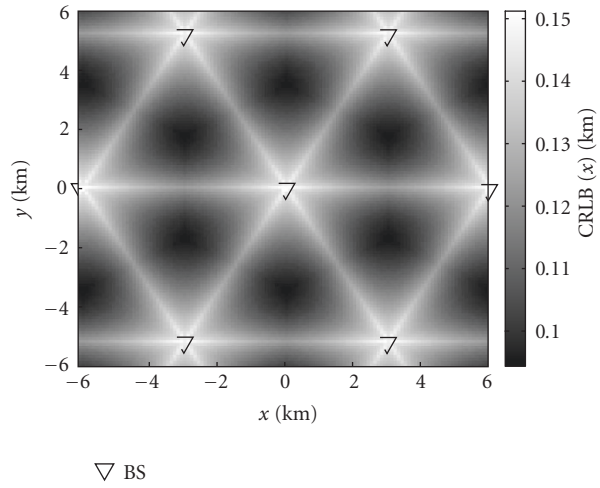

Figure 7: CRLB($\mathbf{x}$) for $\sigma_n = 0.1$ km, $R = 3$ km, $N_{\text{BS}} = 3$.

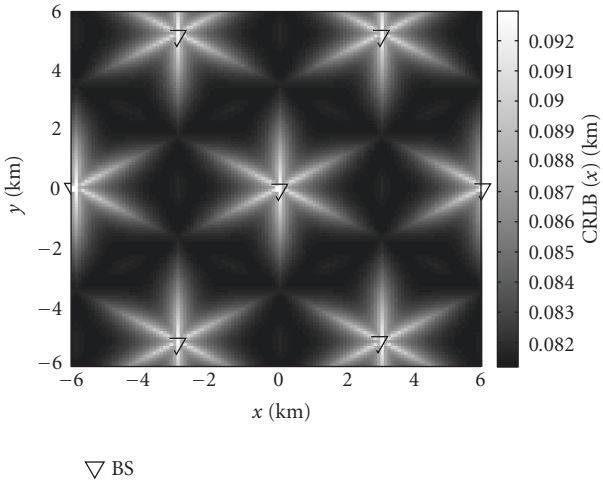

Figure 8: CRLB($\mathbf{x}$) for $\sigma_n = 0.1$ km, $R = 3$ km, $N_{\text{BS}} = 4$.

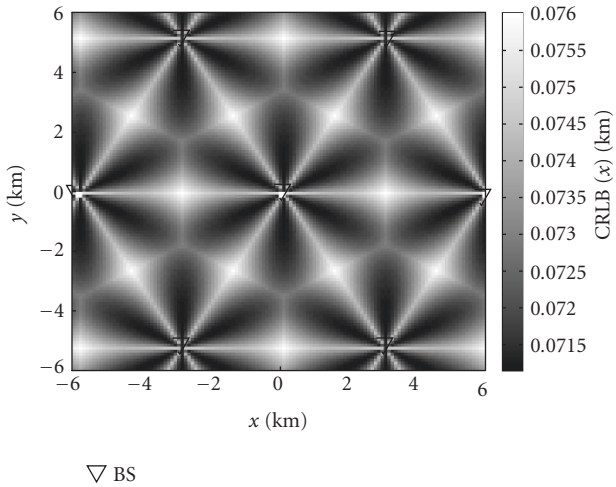

Figure 9: CRLB($\mathbf{x}$) for $\sigma_n = 0.1$ km, $R = 3$ km, $N_{BS} = 5$.

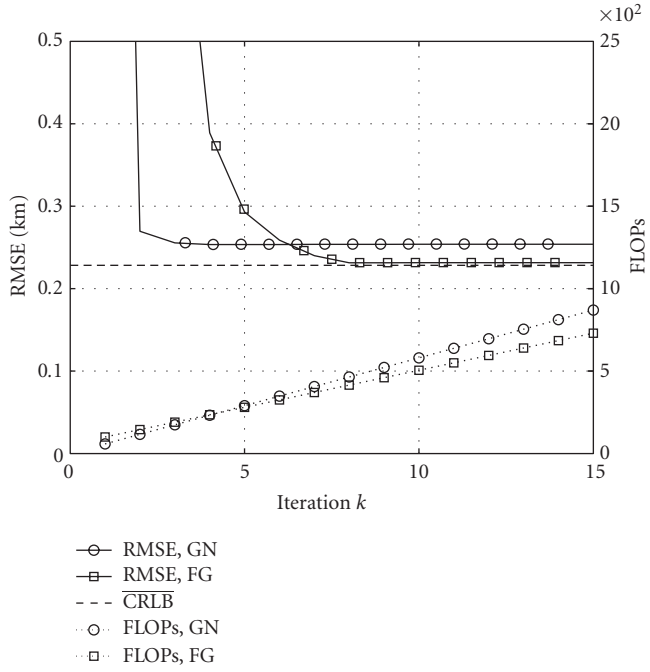

Figure 10: RMSE and FLOPs versus iterations for $\sigma_n = 0.2$ km, $R = 3$ km, $N_{\text{BS}} = 3$.

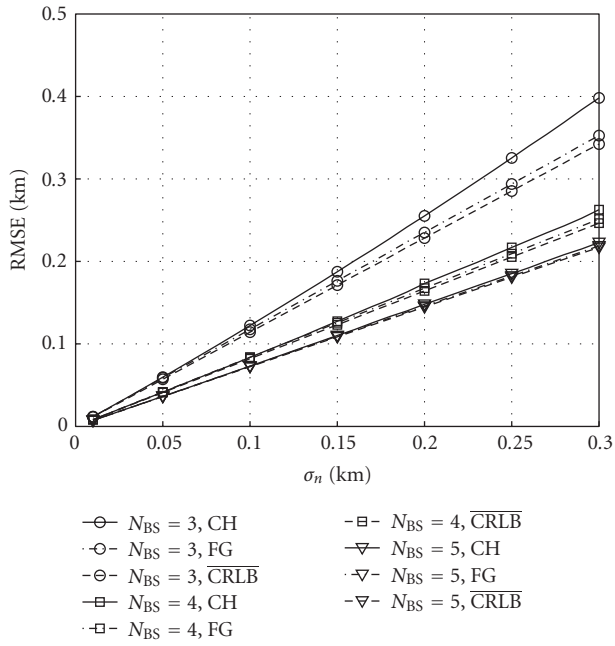

Figure 11: RMSE versus σ_n for FG and CH algorithms, $R = 3$ km.

seen that the deviation from $\overline{\text{CRLB}}$ is very small, even for $N_{\text{BS}} = 3$ and high noise power. To get a better assessment for the performance of the iterative FG algorithm, the results are also compared with a noniterative solution which is based on a method invented by Chan and Ho [17]. It is a three-step procedure extending the spherical interpolation

method [20] and achieving $\overline{\text{CRLB}}$ for low noise power, but with restricted accuracy for higher noise power. The number of required FLOPs is similar compared to the FLOPs for FG with $K_{\text{FG}} = 8$, but we can observe that the performance of the Chan-Ho (CH) method—especially in the most interesting case of $N_{\text{BS}} = 3$—is considerably worse.

We are also interested in CDFs for investigating the performance of the algorithms in general cellular networks. In Figure 12, the performance of GN, FG, and ML as the reference bound is analyzed for $N_{\text{BS}} = 4$ and different values for the noise power σ_n. Note that the quality of the initial value strongly depends on the different MS positions in the cellular network. The estimation error is defined as

$$\varepsilon = ||\mathbf{x} - \hat{\mathbf{x}}||_2, \tag{70}$$

where $\hat{\mathbf{x}}$ is again the final estimate of the algorithms after convergence. The CDF shows the probability that the estimation error ε is below a fixed value ε_{err}, averaged over several MS positions and noise realizations. We observe that FG outperforms the standard GN algorithm for the complete range. However, there is still a gap between FG and the ML bound. This can be explained by observing the CRLB plots (e.g., Figure 8) with the geometric constellations, bad initial values for certain MS positions, and the cycles in the FG. However, the performance difference between GN and ML bounds is much bigger than between FG and ML bounds. Hence, the FG can also in terms of CDFs be seen as more robust against bad geometric constellations and bad inaccurate initial values. Additionally, the FCC rule for emergency calls is shown in Figure 12 (dotted lines). According to the FCC requirements for locating emergency callers, 67% of the positions have to be estimated with an error which is smaller than 0.1 km for network initiated positioning. We see that GN is not suitable to achieve this requirement for $\sigma_n = 0.1$ km, whereas FG fulfils the FCC rule for this scenario.

Figure 13 shows the performance of the FG algorithm in dependence on the number of used BSs for $\sigma_n = 0.1$ km. Clearly, for increasing N_{BS}, also the performance improves. Additionally, the difference between FG and ML gets smaller for increasing N_{BS}. Note that for the simulated scenario, at least $N_{\text{BS}} = 4$ BSs are required to fulfil the FCC requirement.

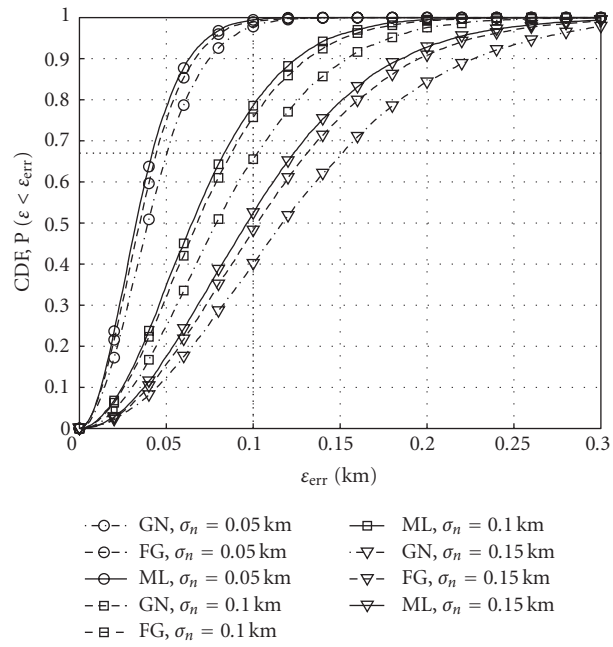

Figure 12: CDF for different algorithms, $R = 3$ km, $N_{\text{BS}} = 4$.

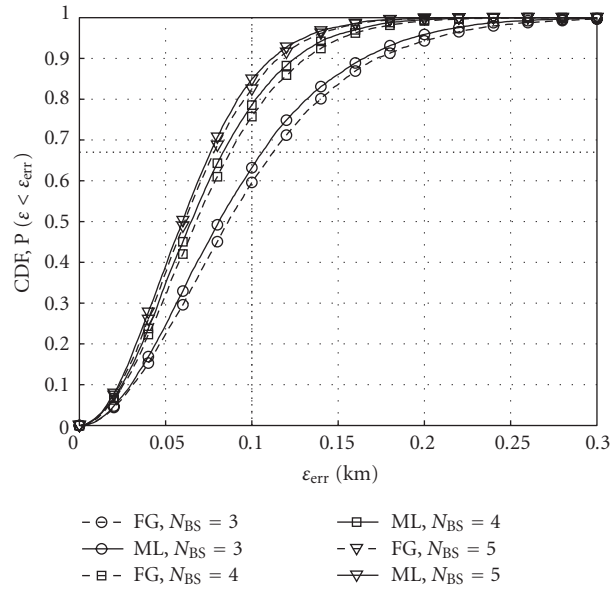

Figure 13: CDF for different numbers of BSs, $R = 3$ km, $\sigma_n = 0.1$ km.

5. CONCLUSIONS

In this paper, we analyzed the mobile station positioning performance in wireless cellular networks using time difference of arrival measurements in a new factor graphs framework. In this scenario, the standard Gauss-Newton algorithm—with similar computational complexity properties—diverges for inaccurate initial values and bad geometric conditions. To avoid these drawbacks, we propose to use a more robust time difference of arrival positioning algorithm based on factor graphs. Simulation results in terms of root-mean-square errors and cumulative density functions show that this method is suitable to estimate the mobile station location with high accuracy and moderate complexity. The proposed method is very close to the Cramer-Rao lower bound and outperforms also the noniterative Chan-Ho algorithm. Furthermore, the performance difference between the factor graphs approach and the optimum—but computational prohibitive—maximum-likelihood solution is very small for various parameters, and thus the proposed algorithm allows the adherence to the FCC emergency call requirements over a more extended range.

ACKNOWLEDGMENT

The material in this paper was presented in part at the IEEE Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Helsinki, Finland, September 2006.

REFERENCES

[1] J. J. Caffery, *Wireless Location in CDMA Cellular Radio Systems*, Kluwer Academic Publishers, Boston, Mass, USA, 2000.

[2] A. H. Sayed, A. Tarighat, and N. Khajehnouri, "Network-based wireless location: challenges faced in developing techniques for accurate wireless location information," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 24–40, 2005.

[3] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 41–53, 2005.

[4] Federal Communications Commission (FCC), FCC 99-245: Third Report and Order. October 1999, http://www.fcc.gov/911/enhanced/.

[5] Coordination Group on Access to Location Information for Emergency Services (CGALIES), Final Report: Report on Implementation Issues Related to Access to Location Information by Emergency Services (E112) in the European Union. February 2002, http://www.telematica.de/cgalies/.

[6] B. W. Parkinson and J. J. Spilker Jr., *Global Positioning System: Theory and Applications, Volume 1*, vol. 163 of *Progress in Astronautics and Aeronautics*, American Institute of Aeronautics & Astronautics, Reston, Va, USA, 1996.

[7] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements and Performance*, Ganga-Jamuna Press, Lincoln, Mass, USA, 2004.

[8] R. Ercek, P. De Doncker, and F. Grenez, "Study of pseudorange error due to non-line-of-sight-multipath in urban canyons," in *Proceedings of the 18th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS '05)*, pp. 1083–1094, Long Beach, Calif, USA, September 2005.

[9] Y. Zhao, "Standardization of mobile phone positioning for 3G systems," *IEEE Communications Magazine*, vol. 40, no. 7, pp. 108–116, 2002.

[10] IST-2003-507581, WINNER Project. https://www.ist-winner.org/.

[11] IST-2003-507581, WINNER Deliverable D4.8.1: WINNER II Intramode and Intermode Cooperation Schemes Definition. June 2006, https://www.ist-winner.org/.

[12] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice-Hall, Upper Saddle River, NJ, USA, 1993.

[13] C. Mensing and S. Plass, "Positioning algorithms for cellular networks using TDoA," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '06)*, vol. 4, pp. 513–516, Toulouse, France, May 2006.

[14] J.-C. Chen, C.-S. Maa, Y.-C. Wang, and J.-T. Chen, "Mobile position location using factor graphs," *IEEE Communications Letters*, vol. 7, no. 9, pp. 431–433, 2003.

[15] J.-C. Chen, Y.-C. Wang, C.-S. Maa, and J.-T. Chen, "Network-side mobile position location using factor graphs," *IEEE Transactions on Wireless Communications*, vol. 5, no. 10, pp. 2696–2704, 2006.

[16] J.-C. Chen, P. Ting, C.-S. Maa, and J.-T. Chen, "Wireless geolocation with TOA/AOA measurements using factor graph and sum-product algorithm," in *Proceedings of the 60th IEEE Vehicular Technology Conference (VTC '04)*, vol. 5, pp. 3526–3529, Los Angeles, Calif, USA, September 2004.

[17] Y. T. Chan and K. C. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Transactions on Signal Processing*, vol. 42, no. 8, pp. 1905–1915, 1994.

[18] W. H. Foy, "Position-location solutions by Taylor-series estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 12, no. 2, pp. 187–194, 1976.

[19] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[20] J. O. Smith and J. S. Abel, "The spherical interpolation method of source localization," *IEEE Journal of Oceanic Engineering*, vol. 12, no. 1, pp. 246–252, 1987.

Christian Mensing studied electrical engineering from 1999 to 2005 at Munich University of Technology (TUM), Germany, with main topics of signal processing and high-frequency technology. He received the B.S., Dipl.-Ing., and M.S. degrees from TUM in 2002, 2004, and 2005, respectively. During his M.S. thesis, he joined the Swiss Federal Institute of Technology Zurich (ETH), Switzerland. He is currently working towards his Ph.D. degree at the Institute of Communications and Navigation of the German Aerospace Center (DLR), Germany. His research interests include location strategies in cellular networks and satellite navigation systems, and efficient iterative detection techniques.

Simon Plass studied at the University of Ulm, Germany, and joined the Oregon State University in Corvallis, Ore, USA, for the academic year 2000. In 2003, he received the Dipl.-Ing. degree from the University of Ulm, Germany. Simon is with the Institute of Communications and Navigation at the German Aerospace Center (DLR), Oberpfaffenhofen, Germany, since 2003. His current interests are cellular wireless communication systems with special emphasis on multicarrier spread-spectrum systems. He is coeditor of *Multi-Carrier Spread Spectrum 2007 (Springer, 2007)*.