11

Sensor Fusion for Position Estimation in Networked Systems

Giuseppe C. Calafiore, Luca Carlone and Mingzhu Wei
Politecnico di Torino
Italy

1. Introduction

Recent advances in wireless communication have enabled the diffusion of networked systems whose capability of acquiring information and acting on wide areas, in a decentralized and autonomous way, represents an attractive peculiarity for many military and civil applications. Sensor networks are probably the best known example of such systems: cost reduction in producing smart sensors has allowed the deployment of constellations of low-cost low-power interconnected nodes, able to sense the environment, perform simple computation and communicate within a given range (Akyildiz et al., 2002). Another example is mobile robotics, whose development has further stressed the importance of distributed control and cooperative task management in formations of agents (Siciliano & Khatib, 2008). A non-exhaustive list of emerging applications of networked systems encompasses target tracking, environmental monitoring, smart buildings surveillance and supervision, water quality and bush fire surveying (Martinez & Bullo, 2006).
The intrinsically distributed nature of measurements acquired by the nodes requires the system to perform a fusion of sensor perceptions in order to obtain relevant information from the environment in which the system is deployed. This is the case of environmental monitoring, in which the nodes may measure the trend of variables of interest over a geographic region, in order to give a coherent overview on the scenario of observation. As in this last example, most of the mentioned fields of application require that each node has precise knowledge of its geometric position for correctly performing information fusion, since actions and observations are location-dependent. Other cases in which it is necessary to associate a position to each node are formation control, which is based on the knowledge of agent positions, and location aware routing, which benefits from the position information for optimizing the flow of data through the network, to mention but a few.
In this chapter we discuss the problem of *network localization*, that is the estimation of node positions from internodal measurements, focusing on the case of pairwise distance measurements. In Section 2 the estimation problem is first introduced, reporting the related literature on the topic. In Section 2.1 we consider the case of localization from range-only measurements, whereas in Section 3 we formalize the estimation problem at hand. Five approaches for solving network localization are extensively discussed in Section 4, where we report the theoretical basis of each technique, the corresponding convergence properties and numerical experiments in realistic simulation setups. The first three localization methods, namely a *gradient-based method*, a *Gauss-Newton* approach and a *trust region* method are local, since

they require a reasonable initial guess on node position to successfully estimate the actual network configuration. We then present two global techniques, respectively a *global continuation* approach and a technique based on s*emidefinite programming* (SDP), which are demonstrated, under suitable conditions, to retrieve the actual configuration, regardless the available prior knowledge on node positions. Several comparative results are presented in Sections 5 and 6. A brief discussion on distributed localization techniques is reported in Section 7 and conclusions are draws in Section 8.

2. Network Localization

When dealing with a network with a large number of nodes a manual configuration of node positions during system set up, when possible, is an expensive and time consuming task. Moreover, in many applications, such as mobile robotics, nodes can move autonomously, thus positions need be tracked as time evolves. A possible solution consists in equipping each node with a GPS sensor, hence allowing the nodes to directly measure their location. Such an approach is often infeasible in terms of cost, weight burden, power consumption, or when the network is deployed in GPS-denied areas. As the above mentioned factors could be technological barriers, a wide variety of solutions for computing node locations through effective and efficient procedures was proposed in the last decade. The so-called *indirects methods* are finalized at determining absolute node positions (with respect to a local or global reference frame) from partial relative measurements between nodes, that is, each node may measure the relative position (angle and distance, angle only or distance only) from a set of *neighbor* nodes, and the global absolute positions of all nodes need be retrieved. This problem is generically known as *network localization*.
If all relative measurements are gathered to some "central elaboration unit" which performs estimation over the whole network, the corresponding localization technique is said to be *centralized*. This is the approach that one implicitly assumes when writing and solving a problem: all the data that is relevant for the problem description is available to the problem solver. In a *distributed* setup, however, each node communicates only with its *neighbors*, and performs local computations in order to obtain an estimate of its own position. As a consequence, the communication burden is equally spread among the network, the computation is decentralized and entrusted to each agent, improving both efficiency and robustness of the estimation process.
In the most usual situation of planar networks, i.e., networks with nodes displaced in two-dimensional space, three main variations of the localization problem are typically considered in the literature, depending on the type of relative measurements available to the nodes. A first case is when nodes may take noisy measurements of the full relative position (coordinates or, equivalently, range and angle) of neighbors; this setup has been recently surveyed in (Barooah & Hespanha, 2007). The localization problem with full position measurements is a linear estimation problem that can be solved efficiently via a standard least-squares approach, and the networked nature of the problem can also be exploited to devise distributed algorithms (such as the Jacobi algorithm proposed in (Barooah & Hespanha, 2007)).
A second case arises, instead, when only angle measurements between nodes are available. This case, which is often referred to as *bearing localization*, can be attacked via maximum likelihood estimation as described in (Mao et al., 2007). This localization setup was pioneered by Stanfield (Stanfield, 1947), and further studied in (Foy, 1976).
In the last case, which is probably the most common situation in practice, each node can measure only distances from a subset of other nodes in the formation. This setup that we shall

name *range localization*, has quite a long history, dating at least back to the eighties, and it is closely related to the so-called molecule problem studied in molecular biology, see (Hendrickson, 1995). However, it still attracts the attention of the scientific community for its relevance in several applications; moreover, recent works propose innovative and efficient approaches for solving the problem, making the topic an open area of research.

2.1 Range localization

The literature on range-based network localization is heterogeneous and includes different approaches with many recent contributions. Most authors formulated the problem in the form of a minimization over a non-linear and non-convex cost function. A survey on both technological and algorithmic aspects can be found in (Mao et al., 2007). In (Howard et al., 2001) the distance constraints are treated using mass-spring models, hence the authors formulate the network localization problem as a minimization of the energy of the overall mass-spring system. The localization problem has also been solved using suitable non linear optimization techniques, like simulated annealing, see (Kannan et al., 2006). First attempts to reduce the computational effort of optimization by breaking the problem into smaller subproblems traces back to (Hendrickson, 1995), in which a divide-and-conquer algorithm is proposed. Similar considerations are drawn in (Moore et al., 2004), where clustering is applied to the network in order to properly reconstruct network configuration. In (More, 1983) the issue of local minima is alleviated using objective function smoothing. In (Biswas & Ye, 2004) the optimization problem is solved using semidefinite programming (SDP), whereas in (Tseng, 2007) network localization is expressed in the form of second-order cone programming (SOCP); sum of squares (SOS) relaxation is applied in (Nie, 2009). Other approaches are based on coarse distance or mere connectivity measurements, see (Niculescu & Nath, 2001) or (Savarese et al., 2002).
Range localization naturally leads to a strongly NP-hard non-linear and non-convex optimization problem (see (Saxe, 1979)), in which convergence to a global solution cannot in general be guaranteed. Moreover the actual reconstruction of a unique network configuration from range measurements is possible only under particular hypotheses on the topology of the networked formation (graph rigidity, see (Eren et al., 2004)). It is worth noticing that localization in an absolute reference frame requires that a subset of the nodes (*anchor nodes* or *beacons*) already knows its exact location in the external reference frame. Otherwise, localization is possible only up to an arbitrary roto-translation. This latter setup is referred to as *anchor-free* localization; see, e.g., (Priyantha et al., 2003).

Notation

I_n denotes the $n \times n$ identity matrix, $\mathbf{1}_n$ denotes a (column) vector of all ones of dimension n, $\mathbf{0}_n$ denotes a vector of all zeros of dimension n, $e_i \in \mathbb{R}^n$ denotes a vector with all zero entries, except for the i-th position, which is equal to one. We denote with $\lfloor x \rfloor$ the largest integer smaller than or equal to x. Subscripts with dimensions may be omitted when they can be easily inferred from context.

For a matrix X, X_{ij} denotes the element of X in row i and column j, and $X^\top$ denotes the transpose of X. $X > 0$ (resp. $X \geq 0$) denotes a positive (resp. non-negative) matrix, that is a matrix with all positive (resp. non-negative) entries. $\|X\|$ denotes the spectral (maximum singular value) norm of X, or the standard Euclidean norm, in case of vectors. For a square matrix $X \in \mathbb{R}^{n,n}$, we denote with $\sigma(X) = \{\lambda_1(X), \ldots, \lambda_n(X)\}$ the set of eigenvalues, or *spectrum*, of X, and with $\rho(X)$ the spectral radius: $\rho(X) \doteq \max_{i=1,\ldots,n} |\lambda_i(X)|$, where $\lambda_i(X)$, $i = 1, \ldots, n$, are the eigenvalues of X ordered with decreasing modulus, i.e. $\rho(X) = |\lambda_1(X)| \geq |\lambda_2(X)| \geq \cdots | \geq |\lambda_n(X)|$.

3. Problem Statement

We now introduce a formalization of the range-based localization problem. Such model is the basis for the application of the optimization techniques that are presented in the following sections and allows to estimate network configuration from distance measurement.
Let $\mathcal{V} = \{v_1, \ldots, v_n\}$ be a set of n nodes (agents, sensors, robots, vehicles, etc.), and let $\mathcal{P} = \{p_1, \ldots, p_n\}$ denote a corresponding set of positions on the Cartesian plane, where $p_i = [x_i \ y_i]^\top \in \mathbb{R}^2$ are the coordinates of the i-th node. We shall call $\mathcal{P}$ a *configuration* of nodes. Consider a set $\mathcal{E}$ of m distinct unordered pairs $e_1, \ldots, e_m$, where $e_k = (i, j)$, and suppose that we are given a corresponding set of nonnegative scalars $d_1, \ldots, d_m$ having the meaning of distances between node i and j.
We want to determine (if one exists) a node configuration $\{p_1, \ldots, p_n\}$ that matches the given set of internodal distances, i.e. such that

$$\|p_i - p_j\|^2 = d_{ij}^2, \quad \forall \ (i,j) \in \mathcal{E},$$

or, if exact matching is not possible, that minimizes the sum of squared mismatch errors, i.e. such that the cost

$$f = \frac{1}{2} \sum_{(i,j)\in\mathcal{E}} \left(\|p_i - p_j\|^2 - d_{ij}^2 \right)^2 \tag{1}$$

is minimized. When the global minimum of f is zero we say that exact matching is achieved, otherwise no geometric node configuration can exactly match the given range data, and we say that approximate matching is achieved by the optimal configuration.
The structure of the problem can be naturally described using graph formalism: nodes $\{v_1, \ldots, v_n\}$ represent the vertices of a graph $\mathcal{G}$, and pairs of nodes $(i, j) \in \mathcal{E}$ between which the internodal distance is given represent graph edges. The cost function f has thus the meaning of accumulated quadratic distance mismatch error over the graph edges. We observe that in practical applications the distance values d_{ij} come from noisy measurements of actual distances between node pairs in a real and existing configuration of nodes in a network. The purpose of network localization is in this case to estimate the actual node positions from the distance measurements. However, recovery of the true node position from distance measurements is only possible if the underlying graph is *generically globally rigid* (ggr), (Eren et al., 2004). A network is said to be globally rigid if is congruent with any network which shares the same underlying graph and equal corresponding information on distances. Generically global rigidity is a stronger concept that requires the formation to remain globally rigid also up to non trivial flexes. Rigidity properties of a network strongly depends on the so called *Rigidity matrix* $R \in \mathbb{R}^{m\times 2n}$, in which each row is associated to an edge e_{ij}, and the four nonzero entries of the row can be computed as $x_i - x_j, y_i - y_j, x_j - x_i, y_j - y_i$ (with $p_i = [x_i, y_i]^\top$), and are located respectively in column $2i - 1, 2i, 2j - 1, 2j$. In particular a network is globally rigid if R has rank $2n - 3$.
If a planar network is generically globally rigid the objective function in (1) has a unique global minimum, if the positions of at least three non-collinear nodes is known and fixed in advance (anchor nodes), or it has several equivalent global minima corresponding to congruence transformations (roto-translation) of the configuration, if no anchors are specified. If the graph is not ggr, instead, there exist many different geometric configurations (also called *flexes*) that match exactly or approximately the distance data and that correspond to equivalent global minima of the cost f. In this work we are not specifically interested in rigidity conditions that

render the global minimum of f unique. Instead, we focus of numerical techniques to compute *a* global minimum of f, that is one possible configuration that exactly or approximately matches the distance data. Clearly, if the problem data fed to the algorithm correspond to a ggr graph with anchors, then the resulting solution will indeed identify univocally a geometric configuration. Therefore, we here treat the problem in full generality, under no rigidity assumptions. Also, in our approach we treat under the same framework both anchor-based and anchor-free localization problems. In particular, when anchor nodes are specified at fixed positions, we just set the respective node position variables to the given values, and eliminate these variables from the optimization. Therefore, the presence of anchors simply reduces the number of free variables in the optimization.

4. Approaches to Network Localization

In this section we review several techniques for solving network localization from range measurements. The first technique is a simple *gradient algorithm* in which the optimization is performed by iterative steps in the direction of the gradient. This approach is able to find a local minimizer of the objective function and requires only first-order information, making the implementation easy and fast. A critical part of the gradient algorithm is the computation of a suitable stepsize. Exact line search prescribes to compute the stepsize by solving a unidimensional optimization problem, hence involving further computational effort in solving the localization. In this context we recall a simple and effective alternative for computing the stepsize, called Barzilai-Borwein stepsize from the name of the authors that proposed it in (Barzilai & Borwein, 1988).
The second technique is a Gauss-Newton (or *iterative least-squares*) approach which is successfully employed in several examples of range localization. We will show how iterative least-squares method converges to the global optimum only in case the initial guess for optimization is reasonably close to the actual configuration. Otherwise the algorithm is only able to retrieve a configuration that corresponds to a local optimum of the objective function. It is worth noticing that, apart from the previous consideration, the algorithm can provide a fast method for obtaining a local solution of the problem.
The third technique is a *trust-region* method which is based on the iterative minimization of a convex approximation of the cost function. The underlying idea is similar to the iterative least-squares: at each step the optimization is performed over a quadratic function which locally resemble the behavior of the objective function. The minimizer of the quadratic approximation is searched over a trust region (a suitable neighborhood of the current point), hence if the approximated solution can assure an appropriate decrease of the objective function the trust region is expanded, otherwise it is contracted. The higher order approximation of the objective function allows trust region to enhance convergence properties, expanding the domain of application of the technique. The improved convergence comes at the price of numerical efficiency, although the trust region method provides a good trade-off between numerical efficiency and global convergence.
In the chapter we further present another solution to the range localization, which is named *global continuation*. This technique was firstly introduced for determining protein structure and for the interpretation of the NMR (Nuclear Magnetic Resonance) data. Global continuation method is based on the idea of iterative smoothing the original cost function into a function that has fewer local minima. Applying a mathematical tool known as *Gaussian transform* the objective function is converted into a convex function and a *smoothing parameter* controls

how much the initial function changes in the transformation. For large values of the smoothing parameter the transformed function is convex, whereas smaller values correspond to less smoothed functions. When the parameter is zero the original cost function is recovered. The result is that the initial smoothing succeeds in moving the initial guess closer to the global optimum of the objective function, then a decreasing sequence of smoothing parameters assures the method to reach the global minimum of the original function. According to the previous considerations the method guarantees the convergence to the global optimum with high probability regardless the initial guess of the optimization. In the chapter it is shown how the robustness of the approach implies a further computation effort which may be unsustainable for nodes with limited computational resources.
Finally we describe a technique which has recently attracted the attention of the research community. The approach, whose first contributions can be found in (Doherty et al., 2001) and (Biswas & Ye, 2004), is based on a relaxation of the original optimization problem and solved using *semidefinite programming* (SDP). This technique is the most computational demanding with respect to the previous approaches, although distributed techniques can be implemented to spread the computational burden on several nodes.
These centralized approaches for minimizing the cost (1) work iteratively from a starting initial guess. As mentioned above the gradient method, the Gauss-Newton approach, the trust region method are local, hence the initial guess plays a fundamental role in the solution of the problem: such techniques may fail to converge to the global optimum, if the initial guess is not close enough to the global solution. In Figure 1 we report an example of node configuration and a possible initial guess for optimization. The Global Continuation method employs iteratively a local approach on a smoothed objective function and this allows the solution to be resilient on perturbations of the initial guess. Finally the Semi-definite Programming approach is proved to retrieve the correct network configuration in the case of exact distance measurements, although it can be inaccurate in the practical case of noisy measurements. The

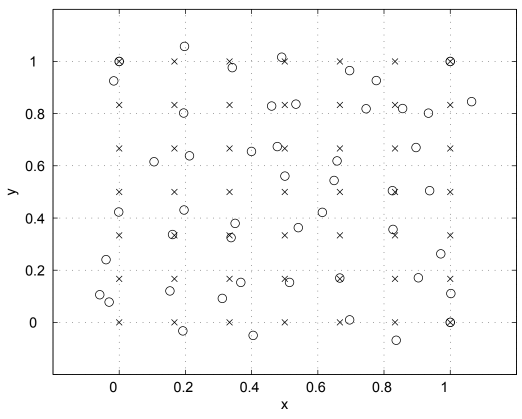


Fig. 1. Actual node configuration (circles) and initial position guess (asterisks).

minimization objective (1) can be rewritten as

$$f(p) = \frac{1}{2} \sum_{(i,j)\in\mathcal{E}} g_{ij}^2(p), \quad g_{ij}(p) \doteq \|p_i - p_j\|^2 - d_{ij}^2, \tag{2}$$

and we let $p^{(0)}$ denote the vector of initial position estimates. We next describe the five centralized methods to determine a minimum of the cost function, starting from $p^{(0)}$.

4.1 A gradient-based method

The most basic iterative algorithm for finding a local minimizer of $f(p)$ is the so called *gradient method* (GM). Let $p^{(\tau)}$ be the configuration computed by the algorithm at iteration τ, being $p^{(0)}$ the given initial configuration: at each iteration the solution is updated according to the rule

$$p^{(\tau+1)} = p^{(\tau)} - \alpha_\tau \nabla f(p^{(\tau)}), \tag{3}$$

where α_τ is the step length, which may be computed at each iteration via exact or approximate line search, and where

$$\nabla f(p) \quad = \sum_{(i,j)\in\mathcal{E}} g_{ij}(p)\nabla g_{ij}(p) \tag{4}$$

where gradient ∇g_{ij} is a row vector of n blocks, with each block composed of two entries, thus $2n$ entries in total, and with the only non-zero terms corresponding to the blocks in position i and j:

$$\nabla g_{ij}(p) \quad = \quad 2[\mathbf{0}_2^\top \ \cdots \ \mathbf{0}_2^\top \ (p_i - p_j)^\top \ \mathbf{0}_2^\top \ \cdots \ \mathbf{0}_2^\top \ (p_j - p_i)^\top \ \mathbf{0}_2^\top \ \cdots \ \mathbf{0}_2^\top].$$

The gradient method is guaranteed to converge to a local minimizer whenever $\{p : f(p) \leq f(p^{(0)})\}$ is bounded and the step lengths satisfy the Wolfe conditions, see, e.g., (Nocedal & Wright, 2006). Although the rate of convergence of the method can be poor, we are interested in this method here since it requires first-order only information (no Hessian needs be computed) and it is, in the specific case at hand, directly amenable to distributed implementation, as discussed in Section 7.

4.1.1 The Barzilai and Borwein scheme

A critical part of the gradient algorithm is the computation of suitable stepsizes α_τ. Exact line search prescribes to compute the stepsize by solving the unidimensional optimization problem

$$\min_\alpha f(p^{(\tau)} - \alpha\nabla f(p^{(\tau)})).$$

Determining the optimal α can however be costly in terms of evaluations of objective and gradient. Moreover, an approach based on exact or approximate line search is not suitable for the decentralized implementation. Barzilai and Borwein (BB) in (Barzilai & Borwein, 1988) proposed an alternative simple and effective technique for selection of the step size, which requires few storage and inexpensive computations. The BB approach prescribes to compute the step size according to the formula

$$\alpha_\tau = \frac{\|p^{(\tau)} - p^{(\tau-1)}\|^2}{(p^{(\tau)} - p^{(\tau-1)})^\top(\nabla f(p^{(\tau)}) - \nabla f(p^{(\tau-1)}))}, \tag{5}$$

hence no line searches or matrix computations are required to determine α_τ. In the rest of the chapter the BB stepsize will be employed for solving the network localization with the gradient method.

4.1.2 Numerical experiments and convergence results

In this section we discuss some numerical experiments that show interesting properties of the gradient-based localization approach.

We first focus on convergence results in the case of exact distance measurements. In the following tests we use generically globally rigid (**ggr**) graphs with n nodes. Hence, by choosing at least three non colinear anchor nodes, the global solution of the localization problem is unique and defines the corresponding geometric configuration of the nodes. One approach to build a **ggr** realization of the networked system is reported in (Eren et al., 2004), and summarized in the following procedure: consider at least 3 non collinear anchor nodes on the plane, then sequentially add new nodes, each one connected with at least 3 anchors or previously inserted nodes. The obtained network is called a *trilateration graph* and it is guaranteed to be **ggr**, see Theorem 8 of (Eren et al., 2004). An example of trilateration graph is reported in Figure 2(a). This technique is fast and easy to implement, however it does not consider that, in

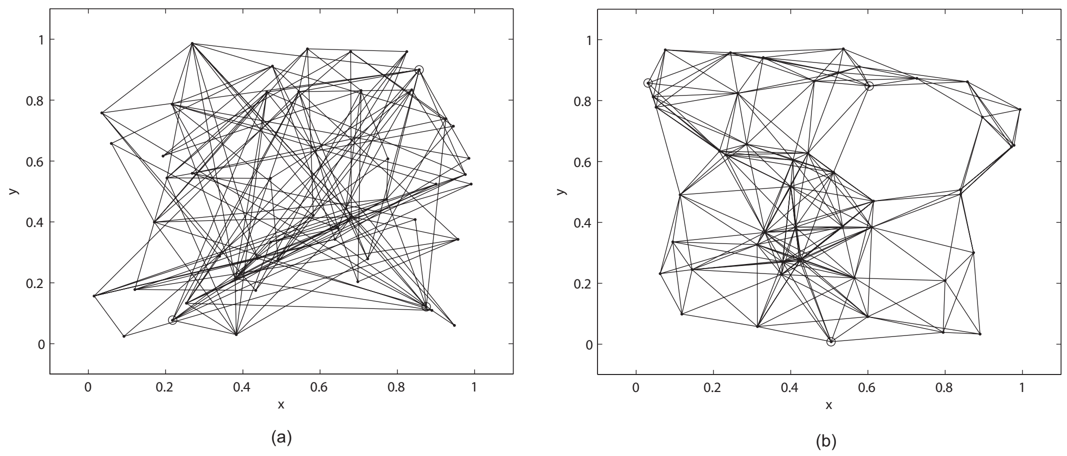


Fig. 2. (a) Example of trilateration graph with nodes in the unit square, $[0,1]^2$; (b) Example of geometric random graph with nodes in the unit square.

practical setups, the *sensing radius* of each node is limited, i.e. edges in the graph may appear only between nodes whose distance is less than the sensing range R. In order to work on more realistic graphs in the numerical tests, we hence use *random geometric graphs*, that are graphs in which nodes are deployed at random in the unit square $[0,1]^2$, and an edge exists between a pair of nodes if and only if their geometrical distance is smaller than R. It has been proved in (Eren et al., 2004) that if $R > 2\sqrt{2}\sqrt{\frac{\log(n)}{n}}$, the graphs produced by the previous technique are **ggr** with high probability. An example of geometric graph with $R = 0.3$ and $n = 50$ is shown in Figure 2(b).

We consider the configuration generated as previously described as the "true" configuration (which is of course unknown in practice), and then, we use the distance measurements from this configuration as the data for the numerical tests. Hence the global optimum of the objective function is expected to correspond to the value zero of the objective function. Convergence properties of the gradient method are evaluated under the settings mentioned above. According to (Moré & Wu, 1997), we consider $p_i^*, i = 1, 2, ..., n$ a solution to the network localization problem, i.e., the gradient algorithm successfully attained the global optimum of the objective function, if it satisfies:

$$| \|p_i - p_j\| - d_{ij} | \leq \varepsilon, \quad (i,j) \in \mathcal{E} \tag{6}$$

where ε is a given tolerance. The technique is local hence it is expected to be influenced

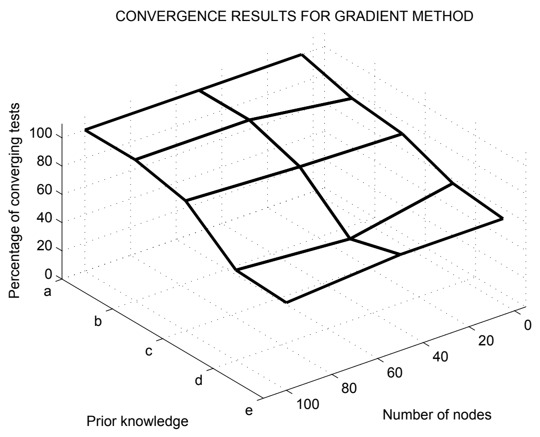


Fig. 3. Percentage of convergence test depending on network size and goodness of initial guess for the GM approach.

by the initial guess for the optimization. In particular, we considered five levels of a-priori knowledge on the configuration:

a) Good prior knowledge: initial guess for the algorithms is drawn from a multivariate Normal distribution centered at the true node positions, with standard deviation $\sigma_p = 0.1$;

b) Initial guess is drawn from a multivariate Normal distribution with $\sigma_p = 0.5$;

c) Bad prior knowledge: Initial guess is drawn from a multivariate Normal distribution with $\sigma_p = 1$;

d) Only the area where nodes are deployed is known: initial guess is drawn uniformly over the unit square;

e) No prior information is available: initial guess is drawn randomly around the origin of the reference frame.

In Figure 3 we report the percentage of test in which convergence is observed for different network sizes and different initial guess on non-anchors position (for each setting we performed 100 simulation runs). The gradient method shows high percentage of convergence when good prior knowledge is available.
The second test is instead related to the localization performance in function of the number of anchors in the network. We consider a realistic setup in which there are 50 non-anchor nodes and the number of anchors ranges from 3 to 10, displaced in the unit square. Two nodes are connected by an edge if their distance is smaller than 0.3 and distance measurement are affected by noise in the form:

$$d_{ij} = \tilde{d}_{ij} + \epsilon_d \ \ \forall \ \ (i,j) \in \mathcal{E} \tag{7}$$

where $\tilde{d}_{ij}$ is the true distance among node i and node j, d_{ij} is the corresponding measured quantity and ϵ_d is a zero mean white noise with standard deviation σ_d. In the following test we consider $\sigma_d = 5 \cdot 10^{-3}$. In order to measure the localization effectiveness, we define the *node positioning error* ϕ_i^* at node i as the Euclidean distance between the estimated position p_i^*

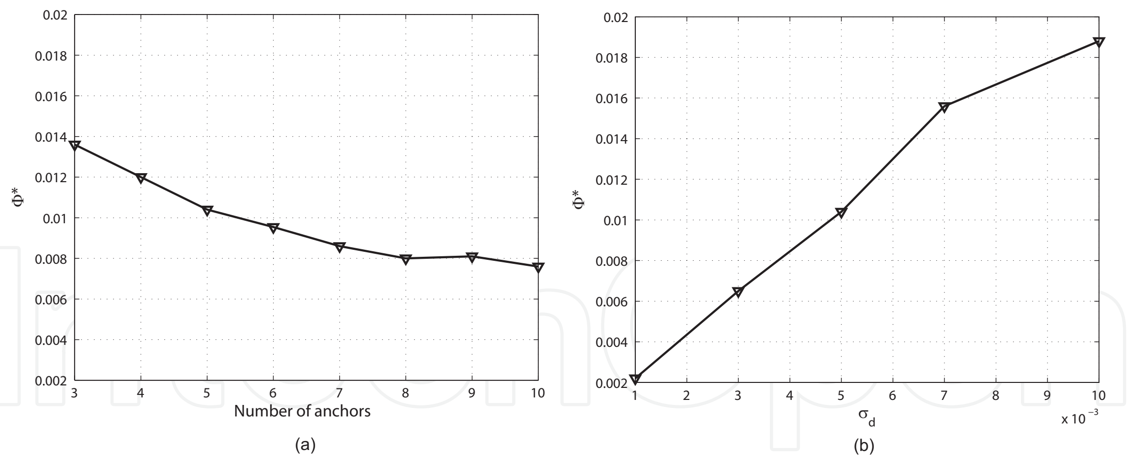


Fig. 4. (a) Localization error for different numbers of anchor nodes, using gradient method; (b) Localization error for different standard deviation of distance measurement noise.

and the true position p_i of the node. The *localization error* Φ^* is further defined as the mean value of the local positioning errors of all the nodes in the network:

$$\Phi^* = \frac{1}{n}\sum_{i=1}^{n} \|p_i - p_i^*\|.$$

It can be seen from Figure 4(a) that the localization error shows low sensitivity on the tested number of anchors, and the downward slope of the curve is not remarked (see tests on SDP, Section 4.5.3, for comparison).
The third test is aimed at studying the localization error for different standard deviations of the distance noise σ_d. The results, considering 3 anchor nodes, are reported in Figure 4(b). It is worth noting that the statistics about the localization error are performed assuming convergence to the global optimum of the technique, hence a good initial guess was used for optimization in the second and third test. In this way we can disambiguate the effects of convergence (Figure 3), from the effect of distance noise propagation (Figures 4(a) and 4(b)).

4.2 Gauss-Newton method

We next discuss a *Gauss-Newton* (GN) approach based on successive linearization of the component costs and *least-squares* iterates.
At each iteration τ of this method, we linearize $g_{ij}(p)$ around the current solution $p^{(\tau)}$, obtaining

$$g_{ij}(p) \simeq g_{ij}(p^{(\tau)}) + \nabla g_{ij}(p^{(\tau)})(p - p^{(\tau)}). \tag{8}$$

Stacking all g_{ij} elements in vector g, in lexicographical order, we have that

$$g(p) \simeq g(p^{(\tau)}) + R(p^{(\tau)})\delta_p(\tau),$$

where $\delta_p(\tau) \doteq p - p^{(\tau)}$, and

$$R(p) \doteq \begin{bmatrix} \nabla g_{i_1 j_1}(p) \\ \vdots \\ \nabla g_{i_m j_m}(p) \end{bmatrix} \in \mathbb{R}^{m,2n}, \tag{9}$$

where m is the number of node pairs among which a relative distance measurement exists. Matrix R is usually known as the *rigidity matrix* of the configuration, see (Eren et al., 2004). Using the approximation in (8), we thus have that

$$\begin{aligned} f(p) &\simeq \frac{1}{2}\sum_{(i,j)\in\mathcal{E}}\left(g_{ij}(p^{(\tau)}) + \nabla g_{ij}(p^{(\tau)})\delta_p(\tau)\right)^2 \\ &= \frac{1}{2}\|g(p^{(\tau)}) + R(p^{(\tau)})\delta_p(\tau)\|^2. \end{aligned}$$

The update step is then computed by determining a minimizing solution for the approximated f, which corresponds to the least-squares solution

$$\delta_p^*(\tau) = -R^+(p^{(\tau)})g(p^{(\tau)}),$$

where R^+ denotes the Moore-Penrose pseudo-inverse of R. Thus, the updated configuration is given by

$$p^{(\tau+1)} = p^{(\tau)} - R^+(p^{(\tau)})g(p^{(\tau)}), \tag{10}$$

and the process is repeated until no further decrease is observed in f, that is until the relative decrease $(f^{(\tau)} - f^{(\tau+1)})/f^{(\tau)}$ goes below a given threshold.
Notice that in the case when anchor nodes are present the very same approach can be used, with the only prescription that the columns of R corresponding to anchors need be eliminated. Specifically, if there are $b > 0$ anchor nodes, we define the reduced rigidity matrix $R_r(p^{(\tau)}) \in \mathbb{R}^{m,2(n-b)}$ as the sub-matrix obtained from $R(p^{(\tau)})$ by removing the pairs of columns corresponding to the anchor nodes, and we define the vector of free variables $\tilde{p}$ as the sub-vector of p containing the coordinates of non-anchor nodes (positions of anchors are fixed, and need not be updated). The iteration then becomes

$$\tilde{p}^{(\tau+1)} = \tilde{p}^{(\tau)} - R_r^+(p^{(\tau)})g(p^{(\tau)}).$$

The described iterative technique is a version of the classical Gauss-Newton method, for which convergence to a local minimizer is guaranteed whenever the initial level set $\{p : f(p) \leq f(p^{(0)})\}$ is bounded, and R_r has full rank at all steps; see, e.g., (Nocedal & Wright, 2006).

4.2.1 Numerical experiments and convergence results

We now present the convergence results for the Gauss-Newton approach, according to the simulation setup presented in 4.1.2. As in the previous example, when no prior information is available we build the initial guess for optimization randomly drawing non-anchor nodes around the origin of the reference frame. It is worth noticing that the initial guess cannot be fixed exactly in the origin otherwise the rank loss in the *rigidity matrix* prevents the application of least squares approach. We denote this first setup as (e) in Figure 5. In the cases prior knowledge on the area in which the network is deployed is available, node positions are initialized randomly in the unit square, $[0,1]^2$. This situation is denoted with (d) in Figure 5. Finally the cases labeled with (a), (b), (c) in Fig. 5 correspond to the case the nodes have some prior information on their geometric location, although this information can be accurate (a), not very accurate (b) or inaccurate (c), see Section 4.1.2. The local nature of the approach is remarked by the 3D plot but is this case the region of attraction of the global minimum is

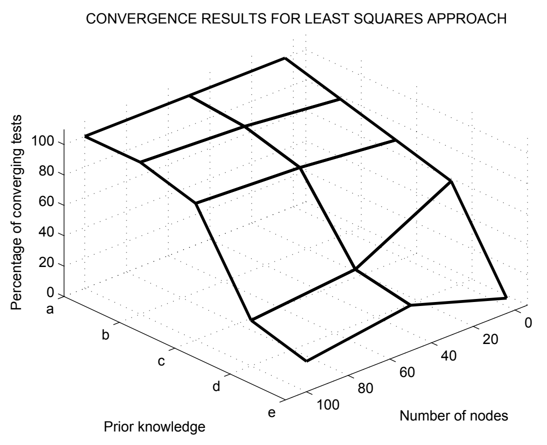


Fig. 5. Percentage of convergence test depending on network size and goodness of initial guess for the GN approach.

smaller and the technique is prone to incur in local minima when starting from poor initial guess. This issue becomes more critical as the number of nodes increases.
We repeated the localization error tests for different measurement noise and number of anchors nodes obtaining exactly the same results as in the gradient-based case. This is however an intuitive result when the initial guess of local techniques is sufficiently close to the global optimum of the objective function: all the techniques simply reaches the same minimum and the localization errors simply depend on the error propagation from distance measurement noise.

4.3 Trust Region approach

The third technique that we examine for application to the problem at hand is a *trust region* (TR) method based on quadratic approximation of the cost function f. At each iteration of this method, the minimizer of the approximated cost function is searched over a suitable neighborhood Δ of the current point (the so-called *trust region*, usually spherical or ellipsoidal). When an adequate decrease of the objective is found in the trust region, the trust region is expanded, otherwise it is contracted, and the process is repeated until no further progress is possible.
The quadratic approximation of f around a current configuration $p^{(\tau)}$ is given by

$$\begin{aligned} f(p) &\simeq q_\tau(p) \\ &\doteq f(p^{(\tau)}) + \nabla f(p^{(\tau)})\delta_p(\tau) + \frac{1}{2}\delta_p^\top(\tau)\nabla^2 f(p^{(\tau)})\delta_p(\tau), \end{aligned}$$

where, using the notation introduced in the previous section,

$$\begin{aligned} \nabla f(p) &= \sum_{(i,j)\in\mathcal{E}} 2g_{ij}(p)\nabla g_{ij}(p) \qquad (11) \\ &= 2g^\top(p)R(p), \end{aligned}$$

and the Hessian matrix $\nabla^2 f \in \mathbb{R}^{2n,2n}$ is given by

$$\nabla^2 f(p) = 2\sum_{(i,j)\in\mathcal{E}} \left(\nabla g_{ij}^\top(p)\nabla g_{ij}(p) + g_v(p)\nabla^2 g_{ij}(p)\right), \qquad (12)$$

where the Hessian matrix $\nabla^2 g_{ij}(p) \in \mathbb{R}^{2n,2n}$ is composed of $n \times n$ blocks of size 2×2: all blocks are zero, except for the four blocks in position $(i,i), (i,j), (j,i), (j,j)$, which are given by

$$\begin{aligned} [\nabla^2 g_{ij}(p)]_{i,i} &= 2I_2, & [\nabla^2 g_{ij}(p)]_{i,j} &= -2I_2, \\ [\nabla^2 g_{ij}(p)]_{j,i} &= -2I_2, & [\nabla^2 g_{ij}(p)]_{j,j} &= 2I_2, \end{aligned}$$

where I_2 denotes the 2×2 identity matrix.
Given a current configuration $p^{(\tau)}$ and trust region Δ_τ, we solve the trust-region subproblem:

$$\min_{\delta_p(\tau) \in \Delta_\tau} \; q_\tau(p).$$

Let $\delta_p^*(\tau)$ be the resulting optimal solution, and let $p^* = p^{(\tau)} + \delta_p^*(\tau)$. Then, we compute the ratio between actual and approximated function decrease:

$$\rho_\tau = \frac{f(p^{(\tau)}) - f(p^*)}{q_\tau(p^{(\tau)}) - q_\tau(p^*)},$$

and update the solution and trust region according to the following rules:

$$p^{(\tau+1)} = \begin{cases} p^{(\tau)} + \delta_p^*(\tau), & \text{if } \rho_\tau > \eta_0 \\ p^{(\tau)}, & \text{if } \rho_\tau \leq \eta_0 \end{cases}$$

$$\xi_{\tau+1} = \begin{cases} \sigma_1 \min\{\|\delta_p^*(\tau)\|, \; \xi_\tau\}, & \text{if } \rho_\tau < \eta_1 \\ \sigma_2 \xi_\tau, & \text{if } \rho_\tau \in [\eta_1, \eta_2) \\ \sigma_3 \xi_\tau, & \text{if } \rho_\tau \geq \eta_2 \end{cases} \tag{13}$$

where $\xi_{\tau+1}$ is the radius of the trust region $\Delta_{\tau+1}$, and $\eta_0 > 0, 0 < \eta_1 < \eta_2 < 1; 0 < \sigma_1 < \sigma_2 < 1 < \sigma_3$ are parameters that have typical values set by experience to $\eta_0 = 10^{-4}, \eta_1 = 0.25, \eta_2 = 0.75; \sigma_1 = 0.25, \sigma_2 = 0.5, \sigma_3 = 4$. The conjugate gradient method is usually employed to solve the trust-region subproblem, see (More, 1983) for further implementation details.

4.3.1 Numerical experiments and convergence results

In this section we report the results on the application of the Trust Region approach to the network localization problem. We now focus on the convergence properties of the approach whereas further comparative experiments are presented in Section 5. The simulation are performed according to the setup described in Section 4.1.2 and 4.2.1. Figure 6 shows the percentage of convergence for different initial guesses of the optimization and for different network sizes. The statistics are performed over 100 simulation runs. The trust region approach provides better convergence properties with respect to Gauss-Newton approach, although also showing degrading performance under scarce prior knowledge. Regarding the sensitivity to the number of anchors and to the measurement noise, the results reported in Section 4.1.2 hold (see also the final remarks of Section 4.2.1).

4.4 Global Continuation approach

The *global continuation* (GC) method is based on the idea of gradually transforming the original objective function into smoother functions having fewer local minima. Following the

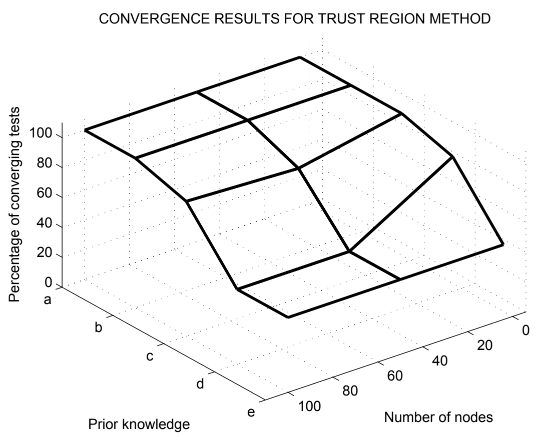


Fig. 6. Percentage of convergence test depending on network size and goodness of initial guess for TR approach.

approach of (More, 1983), the smoothed version of the cost function is obtained by means of the Gaussian transform: For a function $f : \mathbb{R}^n \to \mathbb{R}$ the Gaussian transform is defined as

$$\varphi(x) \doteq \langle f \rangle_\lambda(x) = \frac{1}{\pi^{n/2}} \int_{R^n} f(x + \lambda u) \exp(-\|u\|^2) \mathrm{d}u. \tag{14}$$

Intuitively, $\varphi(x)$ is the average value of $f(z)$ in the neighborhood of x, computed with respect to a Gaussian probability density. The parameter λ controls the degree of smoothing: large λ implies a high degree of smoothing, whereas for $\lambda \to 0$ the original function f is recovered. The Gaussian transform of the cost function in (1) can be computed explicitly, see Theorem 4.3 on (More, 1983).

Proposition 1 (Gaussian transform of localization cost). *Let f be given by (1). Then, the Gaussian transform of f is given by*

$$\varphi_\lambda(p) = f(p) + \gamma + 8\lambda^2 \sum_{(i,j)\in\mathcal{E}} \|p_i - p_j\|^2, \tag{15}$$

where

$$\gamma = 8m\lambda^4 - 4\lambda^2 \sum_{(i,j)\in\mathcal{E}} d_{ij}^2.$$

It is interesting to observe that, for suitably large value of λ, the transformed function $\varphi_\lambda(p)$ is *convex*. This fact is stated in the following proposition.

Proposition 2 (Convexification of localization cost). *Let $f(p)$ be given by (1), and let $\varphi_\lambda(p)$ be the Gaussian transform of $f(p)$. If*

$$\lambda > \frac{1}{2} \max_{(i,j)\in\mathcal{E}} d_{ij} \tag{16}$$

then $\varphi_\lambda(p)$ is convex.

Proof. From (1) and (15) we have that

$$\varphi_\lambda(p) = \gamma + \sum_{(i,j)\in\mathcal{E}} 8\lambda^2 r_{ij}^2(p) + (r_{ij}^2(p) - d_{ij}^2)^2,$$

where we defined $r_{ij}(p) \doteq \|p_i - p_j\|$. Let

$$h_{ij}(r_{ij}) = 8\lambda^2 r_{ij}^2 + (r_{ij}^2 - d_{ij}^2)^2.$$

Then

$$\begin{aligned} h'_{ij} \doteq \frac{\mathrm{d}h_{ij}}{\mathrm{d}r_{ij}} &= 4r_{ij}(r_{ij}^2 - d_{ij}^2 + 4\lambda^2) \\ h''_{ij} \doteq \frac{\mathrm{d}^2 h_{ij}}{\mathrm{d}r_{ij}^2} &= 4(3r_{ij}^2 - d_v^2 + 4\lambda^2). \end{aligned}$$

Note that $h'_{ij} > 0$ if $4\lambda^2 > d_{ij}^2 - r_{ij}^2$, and $h''_{ij} > 0$ if $4\lambda^2 > d_{ij}^2 - 3r_{ij}^2$. Since $r_{ij} \geq 0$ it follows that, for $4\lambda^2 > d_{ij}^2$, both h'_{ij} and h''_{ij} are positive. Therefore, if

$$\lambda > \frac{1}{2} \max_{(i,j)\in\mathcal{E}} d_{ij} \tag{17}$$

then functions h_{ij} are increasing and convex, for all (i, j). Observe next that function $r_{ij}(p)$ is convex in p (it is the norm of an affine function of p), therefore by applying the composition rules to $h_{ij}(r_{ij}(p))$ (see Section 3.2.4 of (Boyd, 2004)), we conclude that this latter function is convex. Convexity of $\varphi_\lambda(p)$ then immediately follows since the sum of convex functions is convex. □

The key idea in the global continuation approach is to define a sequence $\{\lambda_k\}$ decreasing to zero as k increases, and to compute a corresponding sequence of points $\{p^*(\lambda_k)\}$ which are the global minimizers of functions $\varphi_{\lambda_k}(p)$. The following strong result holds.

Proposition 3 (Theorem 5.2 of (More, 1983)). *Let $\{\lambda_k\}$ be a sequence converging to zero, and let $\{p^*(\lambda_k)\}$ be the corresponding sequence of global minimizers of $\varphi_{\lambda_k}(p)$. If $\{p^*(\lambda_k)\} \to p^*$ then p^* is a global minimizer of $f(p)$.*

In practice, we initialize λ to a value λ_1 that guarantees convexity of φ_λ, so that the initial computed point is guaranteed to be the global minimizer of φ_λ. Then, λ is slightly decreased and a new minimizer is computed using the previous point as the starting guess. The process is iterated until $\lambda = 0$, that is until the original cost function f is minimized. Each iteration in this procedure requires the solution of an unconstrained minimization problem, which can be suitably performed using a trust-region method. The various steps of the global continuation method are summarized in the next section.

4.4.1 Global Continuation algorithm

Set the total number M of major iterations. Given an initial guess p_0 for the configuration:

1. Let $k = 1$. Compute a convexifying parameter
$$\lambda_k = \frac{1}{2} \max_{(i,j)\in\mathcal{E}} d_{ij};$$
2. Compute a (hopefully global) minimizer p_k^* of $\varphi_{\lambda_k}(p)$ using a trust-region method with initial guess p_{k-1};
3. If $\lambda_k = 0$, exit and return $p^* = p_k^*$;

4. Let $k = k + 1$. Update λ:
$$\lambda_k = \frac{M-k}{M-1}\lambda_1;$$
5. Go to step 2).

In step 2) of the algorithm, a quadratic approximation of $\varphi_{\lambda_k}(p)$ is needed for the inner iterations of the trust-region method. More precisely, the trust-region algorithm shall work with the following approximation of φ_λ around a current point $\bar{p}$:

$$\begin{aligned}\varphi_\lambda(p) &\simeq q_\lambda(p)\\ &\doteq \varphi_\lambda(\bar{p}) + \nabla\varphi_\lambda(\bar{p})\delta_p + \frac{1}{2}\delta_p^\top \nabla^2\varphi_\lambda(\bar{p})\delta_p,\end{aligned}$$

where $\delta_p = p - \bar{p}$. Due to the additive structure of (15), the gradient and Hessian of φ_λ are computed as follows:

$$\begin{aligned}\nabla\varphi_\lambda(p) &= \nabla f(p) + 8\lambda^2 \sum_{(i,j)\in\mathcal{E}} \nabla g_{ij}(p)\\ \nabla^2\varphi_\lambda(p) &= \nabla^2 f(p) + 8\lambda^2 \sum_{(i,j)\in\mathcal{E}} \nabla^2 g_{ij}(p).\end{aligned}$$

4.4.2 Numerical experiments and convergence results

Repeating the localization test performed for the techniques presented so far, we report the percentage of convergence for the Global Continuation approach in Figure 7. The different levels of prior knowledge are the same as in Sections 4.1.2 and 4.2.1. We choose a number of major (or outer) iterations $M = 11$. The global continuation method, although computationally

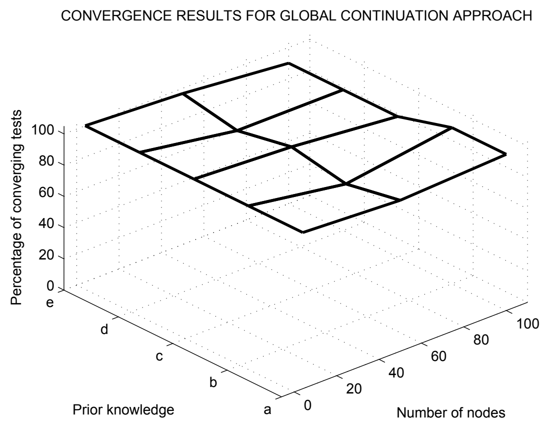


Fig. 7. Percentage of convergence test depending on network size and goodness of initial guess for GC approach.

more intensive than the previous two methods (see Section 5), shows instead a remarkable insensitivity to the initial guess, and therefore it is suitable for applications in which little or no prior knowledge is available. In few cases the number of converging experiments is less than 100%, but this issue can be alleviated by increasing the number of major iterations, hence making a more gradual smoothing. Further results on computational effort required by the technique in reported in the Section 5.

4.5 Semidefinite Programming-based localization

In this section we describe an approach to network localization, which is based on *semidefinite programming* (SDP). First attempts to solve the range localization problem using convex optimization trace back to (Doherty et al., 2001); the authors model some upper bounds on distance measurement as convex constraints, in the form:

$$\|p_i - p_j\| \leq d_{ij}, \quad (i,j) \in \mathcal{E}. \tag{18}$$

The previous convex constraints can only impose internodal distances to be less than a given sensing range or less or equal to a measured distance d_{ij}. However, as stated in Section 3, we want to impose equality constraints in the form:

$$\|p_i - p_j\|^2 = d_{ij}^2, \quad (i,j) \in \mathcal{E}, \tag{19}$$

Such constraints are non convex, and the SDP network localization approach is based on a relaxation of the original problem. If only inequality conditions like (18) are used it is possible to assure good localization accuracy only when non-anchor nodes are in the convex hull of the anchors, whereas these localization approaches tend to perform poorly when anchors are placed in the interior of the network (Biswas et al., 2006).
The rest of this section is structured as follows. The SDP relaxation is presented in Section 4.5.1. Then in Section 4.5.2 some relevant properties of the SDP approach are discussed. In Section 4.5.3 some numerical examples are reported. Finally, a gradient refinement phase is presented in Section 4.5.4, for the purpose of enhancing localization performance when the distance measurements are affected by noise.

4.5.1 Problem relaxation

In the previous sections, we found it convenient to stack the information on node positions, $p_i = [x_i \; y_i]^\top$, $i = 1, \dots, n$, in the column vector $p \doteq [p_1^\top \; p_2^\top \; \cdots \; p_n^\top]^\top \in \mathbb{R}^{2n}$. Here, we modify the notation and pack the variables p_i as follows

$$p \quad = \quad \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \end{bmatrix} = [p_1 \; p_2 \; \dots \; p_n] \in \mathbb{R}^{2\times n},$$

As mentioned in Section 3 if an anchor-based setup is considered the columns of p are simply deleted and the corresponding Cartesian positions are substituted in the problem formulation, with known vectors $a_k \in \mathbb{R}^2$. In the following we recall the relaxation approach of (Biswas et al., 2006) and (Biswas & Ye, 2004).
Starting from equation (19) we can distinguish constraints which are due to measurements between two non-anchor nodes and measurements in which an anchor node is involved. For this purpose we partition the set $\mathcal{E}$ into two sets, respectively called $\mathcal{E}_p$ (including all edges between non-anchor nodes) and $\mathcal{E}_a$ (including all edges incident on one anchor node). We further define $m_p = |\mathcal{E}_p|$ and $m_a = |\mathcal{E}_a|$, where $|S|$ denotes the cardinality of the set S. Therefore the localization problem can be rewritten as:

$$\begin{aligned} \|p_i - p_j\|^2 &= d_{ij}^2, \quad \forall (i,j) \in \mathcal{E}_p \\ \|a_k - p_j\|^2 &= d_{kj}^2, \quad \forall (k,j) \in \mathcal{E}_a \end{aligned}$$

where d_{ij} is the distance measurement between non-anchor nodes i and j and d_{kj} is the distance measurement between non-anchor node j and anchor node k.

If we define the standard unit vector e_i as a column vector of all zeros, except a unit entry in the i-th position, it is possible to write the following equalities:

$$\begin{aligned}
\|p_i - p_j\|^2 &= (e_i - e_j)^\top p^\top p(e_i - e_j), \ \forall (i,j) \ \in \mathcal{E}_p \\
\|a_k - p_j\|^2 &= \begin{pmatrix} a_k \\ -e_j \end{pmatrix}^\top \begin{pmatrix} I_2 & p \\ p^\top & Y \end{pmatrix} \begin{pmatrix} a_k \\ -e_j \end{pmatrix}, \ \forall \ (k,j) \in \mathcal{E}_a, \quad Y = p^\top p,
\end{aligned}$$

Then the matrix form of the localization problem can be rewritten as:

$$\begin{array}{ll}
\textbf{find} & p \in \mathbb{R}^{(2\times n)}, Y \in \mathbb{R}^{(n\times n)} \\
\textbf{s.t.} & (e_i - e_j)^\top Y (e_i - e_j) = d_{ij}^2, \ \forall \ (i,j) \in \mathcal{E}_p \\
& \begin{pmatrix} a_k \\ -e_j \end{pmatrix}^\top \begin{pmatrix} I_2 & p \\ p^T & Y \end{pmatrix} \begin{pmatrix} a_k \\ -e_j \end{pmatrix} = d_{kj}^2, \ \forall \ (k,j) \in \mathcal{E}_a; \\
& Y = p^\top p.
\end{array} \tag{20}$$

Equation (20) can be relaxed to a semidefinite program by simply substituting the constraint $Y = p^\top p$ with $Y \succeq p^\top p$. According to (Boyd, 2004) the previous inequality is equivalent to:

$$Z = \begin{pmatrix} I_2 & p \\ p^\top & Y \end{pmatrix} \succeq 0.$$

Then the relaxed problem (20) can be stated in standard SDP form:

$$\begin{array}{ll}
\textbf{min} & \mathbf{0} \\
\textbf{s.t.} & (1;0;\mathbf{0})^\top Z(1;0;\mathbf{0}) = 1 \\
& (0;1;\mathbf{0})^\top Z(0;1;\mathbf{0}) = 1 \\
& (1;1;\mathbf{0})^\top Z(1;1;\mathbf{0}) = 2 \\
& (0;e_i - e_j)^\top Z(0;e_i - e_j) = d_{ij}^2, \ \forall \ (i,j) \in \mathcal{E}_p, \\
& \begin{pmatrix} a_k \\ -e_j \end{pmatrix}^\top Z \begin{pmatrix} a_k \\ -e_j \end{pmatrix} = d_{kj}^2, \ \forall \ (k,j) \in \mathcal{E}_a, \\
& Z \succeq 0.
\end{array} \tag{21}$$

Problem (21) is a feasibility convex program whose solution can be efficiently retrieved using interior-point algorithms, see (Boyd, 2004). As specified in Section 4.5.2 the approach is proved to attain the actual node position, regardless the initial guess chosen for optimization. It is clear that constraints in (21) are satisfied only if all distance measurements exactly match internodal distances in network configuration. For example, in the ideal case of perfect distance measurements, the optimal solution satisfies all the constraints and corresponds to the actual node configuration. In practical applications, however, the distance measurements are noisy, and, in general, no configuration does exist that satisfies all the imposed constraints. In such a case it is convenient to model the problem so to minimize the error on constraint satisfaction, instead of the stricter feasibility form (21). Hence the objective function can be rewritten as the sum of the error between the measured ranges and the distances between the nodes in the estimated configuration:

$$f_{SDP}(p) = \sum_{(i,j)\in\mathcal{E}_p} \left| \|p_i - p_j\|^2 - d_{ij}^2 \right| + \sum_{(k,j)\in\mathcal{E}_a} \left| \|a_k - p_j\|^2 - d_{kj}^2 \right| \tag{22}$$

It is worth noticing that, if the square of the errors is considered instead of the absolute value, the problem formulation exactly matches the one presented in Section 3. By introducing slack variables u_s and l_s, the corresponding optimization problem can be stated as follows:

$$\begin{array}{ll} \min & \sum_{(i,j)\in\mathcal{E}_p}(u_{ij}+l_{ij}) + \sum_{(k,j)\in\mathcal{E}_a}(u_{kj}+l_{kj}) \\ \mathbf{s.t.} & (1;0;\mathbf{0})^\top Z(1;0;\mathbf{0}) = 1 \\ & (0;1;\mathbf{0})^\top Z(0;1;\mathbf{0}) = 1 \\ & (1;1;\mathbf{0})^\top Z(1;1;\mathbf{0}) = 2 \\ & (0;e_i-e_j)^\top Z(0;e_i-e_j) - u_{ij} + l_{ij} = d_{ij}^2, \;\; \forall\ (i,j)\in\mathcal{E}_p, \\ & \begin{pmatrix} a_k \\ -e_j \end{pmatrix}^\top Z \begin{pmatrix} a_k \\ -e_j \end{pmatrix} - u_{kj} + l_{kj} = d_{kj}^2, \;\; \forall\ (i,j)\in\mathcal{E}_a, \\ & u_{ij}, u_{kj}, l_{ij}, l_{kj} \geq 0, \\ & Z \succeq 0. \end{array} \tag{23}$$

The previous semidefinite convex program allows to efficiently solve the range localization; moreover it has global convergence properties as we describe in the following section.

4.5.2 Analysis of SDP-based network localization

In the ideal situation of agents that can take noiseless measurements of internodal distances, and when the network reconstruction is unique, the SDP approach allows to estimate the exact network configuration, (Biswas et al., 2006), (Biswas & Ye, 2004). The solution of the feasibility problem (21) is then:

$$Z^* = \begin{pmatrix} I_2 & p^* \\ p^{*\top} & Y^* \end{pmatrix}, \quad Y^* = p^{*\top} p^*. \tag{24}$$

Hence the problem with the relaxed condition $Y \succeq p^\top p$ allows to retrieve the solution that satisfies the original problem with the constraint $Y = p^\top p$. Further discussion on conditions that make a network uniquely localizable and their connection with rigidity theory are reported in (So & Ye, 2007).

When dealing with noisy distance measurements, the SDP solution is no longer guaranteed to attain the global minimum of the objective function. In such a case the approach with relaxed constraints may provide inaccurate position estimation. In particular the matrix Y^* may have rank higher than 2, i.e., the estimated solution lies in a higher dimensional space than the planar dimension of the original problem. In (Biswas & Ye, 2004) the higher dimensional solution is simply projected in $\mathbb{R}^2$, but this comes at a price of rounding errors which may influence localization performance. For the purpose of improving position estimation accuracy in Section 4.5.4 we further describe a gradient refinement phase that may be added to the SDP localization.

We further observe that the SDP network localization, as expressed in (21), is able to retrieve the actual network configuration in case we consider a **ggr** graph, i.e., the global optimum of the objective function is unique. If the solution is not unique, the SDP approach returns a central solution that is the mean value of the global optima positions (So & Ye, 2007). We conclude this section with a remark on the complexity of the SDP network localization approach.

Remark 1. *Let $c = m_p + m_a + 3$ be the number of constraints in the SDP formulation of the range localization problem (23) and ε be a positive number. Assume that a ε-solution of (23) is required, that is we are looking for an optimal configuration of the network that corresponds to a value of the objective function that is at most ε above the global minimum. Then the total number of interior-point* *algorithm iterations is smaller than* $\sqrt{n+c}\log\frac{1}{\varepsilon}$*, and the worst-case complexity of the SDP approach is* $O(\sqrt{n+c}(n^3+n^2c+c^3\log\frac{1}{\varepsilon}))$.

4.5.3 Numerical experiments

For numerical experiments we consider a geometric random graph with nodes in the unit square and sensing radius $R = 0.3$. We focus on the noisy case in which distance measurement are affected by noise with standard deviation $\sigma_d = 5 \cdot 10^{-3}$, according to the model (7). The gradient method, Gauss-Newton technique, the trust region approach and the global continuation were verified to be quite insensitive to the number of anchors. The SDP solution in presence of noise, however, is influenced by the number of anchors and on their placement. In order to show the effect of anchors on localization performance we consider a network with 50 non-anchor nodes and we varied the number of anchors from 3 to 10. Results are reported in Figure 8(a). We further tested the effects of anchor placement on localization performance: we

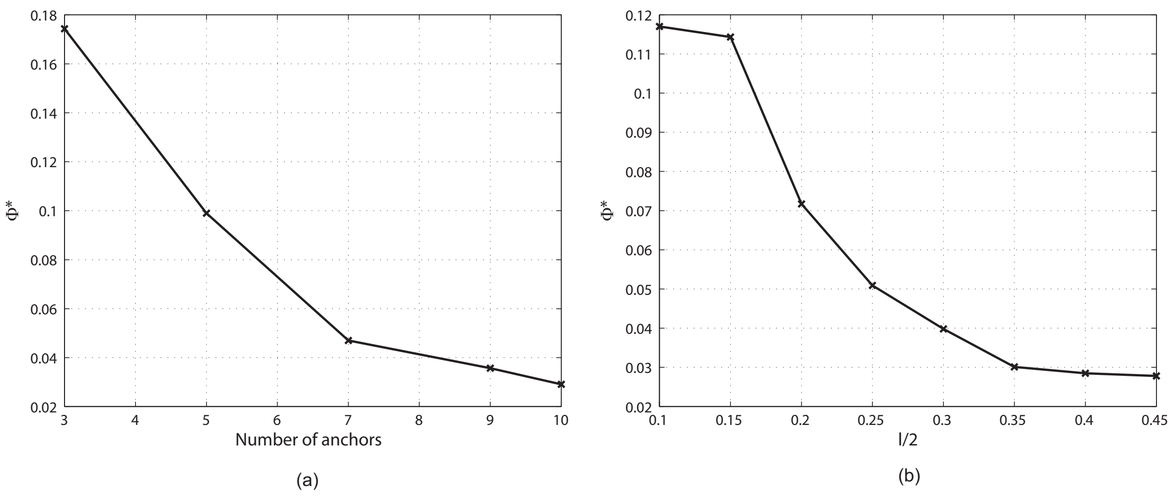


Fig. 8. (a) Localization error for different number of anchor nodes, using SDP approach; (b) Localization error for different anchor placements, using SDP approach. Four anchor nodes are displaced on the vertices of a square centered in [0.5, 0.5] and side l.

consider a network in the unit square, with four anchors disposed on the vertices of a smaller square with side l, centered in $[0.5, 0.5]$. For a small value of l the anchors are in the interior of the network, whereas as l increases the anchors tends to be placed on the boundary of the formation. It is possible to observe that the latter case, i.e., when the non-anchor nodes are in the convex hull of the anchors, the localization error is remarkably reduced, see Figure 8(b).

4.5.4 SDP with local refinement

As mentioned in Section 4.5.1, when distance measurements are affected by some measurement noise, the SDP can be inaccurate. On the other hand the reader can easily realize from the numerical experiments presented so far, that the local approaches assures better accuracy when an accurate initial guess is available. Therefore in a situation in which no a-priori information on node position is available to the problem solver and distance measures are supposed to be noisy, one may take the best of both local and global approaches by subdividing the localization problem into two phases (Biswas et al., 2006): (i) apply the SDP relaxation in order to have a rough estimate of node position; (ii) refine the imprecise estimated configuration with a local approach. For example one can use the gradient method described in Section 4.1, which is simple and has good convergence properties at a time. Moreover the gradient approach can be easily implemented in a distributed fashion, as discussed in Section 7. Numerical results on the SDP with local refinement are presented in Section 6.

5. Performance Evaluation of Local Approaches

We now compare the local techniques presented so far, i.e., the gradient method, the Gauss-Newton approach and the trust region approach. Global continuation is also reported for comparison, although it has global convergence capability. We consider a first setup that exemplifies the case of a network of sensors in which few nodes are equipped with GPS (anchors), whereas others use indirect estimation approaches (like the ones presented so far) for localization. For a numerical example, we considered the case of n nodes located on terrain according to the configuration shown Figure 9(b). This actual configuration should be estimated autonomously by the agents using as prior knowledge a given initial guess configuration, as the one shown in Figure 9(a). Four anchor nodes are selected at the external vertices of the network. Simulations were performed with $n = 49$ nodes. We test the convergence of the tech-

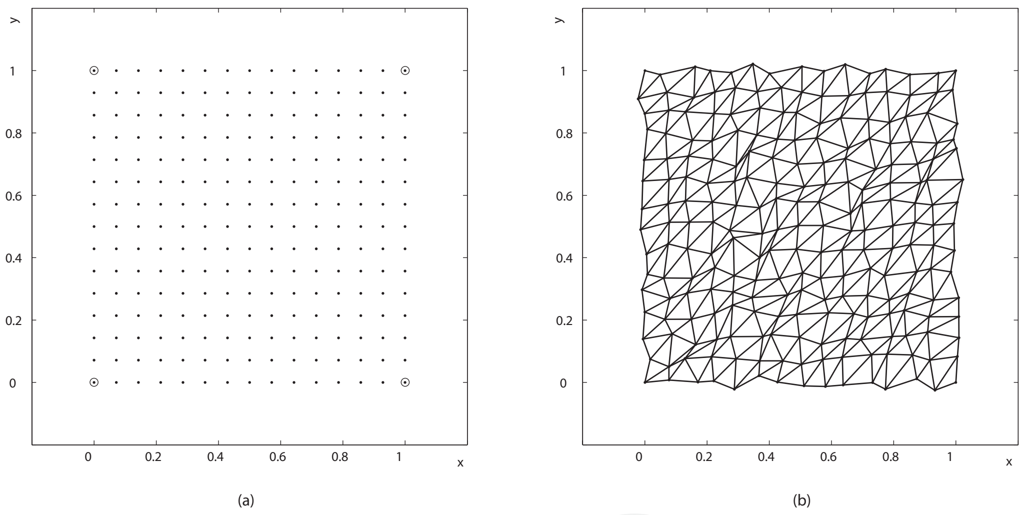


Fig. 9. (a) initial position guess; (b) actual nodes configuration ($n = 225$).

niques, depending on the initial guess. The latter is obtained from the actual configuration by perturbing each node position with a Gaussian noise with covariance matrix $\Sigma_i = \sigma_p^2 I_2$. Figure 10 reports the percentage of converging tests (as defined in Section 4.1.2) over 100 simulation runs. For the same experiment we report the computational effort for the four techniques, see Table 1. The effort is expressed in terms of CPU time required for reaching the termination condition of each algorithm. The tests were performed in Matlab on a MacBook, with 2.26 GHz clock frequency and 4 GB RAM.
It is possible to observe that the Gauss-Newton is the less resilient when the initial guess is not accurate, although it is fast and converges in few iterations. The Gradient method and the trust region approach have similar convergence properties and they require a comparable computational effort: the trust region approach is able to converge in few iterations, since it

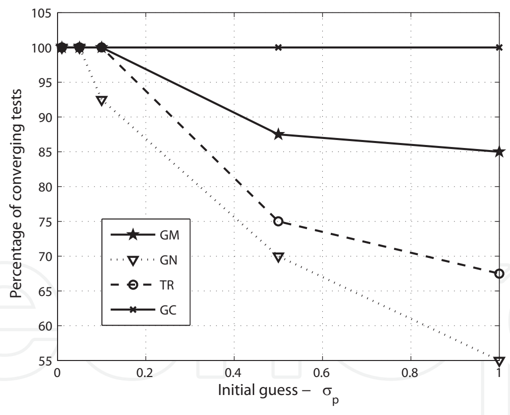


Fig. 10. Percentage of convergence vs. goodness of initial guess.

Table 1. CPU time for gradient method (GM), Gauss-Newton (GN), trust region (TR) and global continuation (GC) approach for different values of σ_p. Time is expressed in seconds.

σ_p	GM	GN	TR	GC
0.01	0.2955	0.0264	0.0751	2.2362
0.05	0.3292	0.0393	0.0635	2.2280
0.1	0.3657	0.0869	0.0930	2.2437
0.5	0.4449	0.7493	0.2316	2.3654
1	0.5217	1.4703	0.3443	2.5524

uses also second order information on the objective function (i.e., the Hessian). The gradient method, however, requires simpler update steps, but this comes at the price at a bigger number of iteration required for the technique to converge. Finally the global continuation converged in all the test performed, whereas the computational effort required is remarkably higher than the other approaches. Table 1 also enlightens how global continuation takes no advantage from good prior knowledge, since the initial smoothing moves the initial guess to the minimizer of the convexified function, regardless the starting guess of the optimization.

6. Localization Performance of Global Approaches

In this section we analyze the localization performance of the global approaches introduced so far, i.e., the global continuation approach and the SDP relaxation. We further compare the SDP with a local refinement, in which a gradient method is added in cascade to the SDP approach, as described in Section 4.5.4. We consider the rigid lattice shown in Figure 9, with $n = 49$ and we report the localization error for different standard deviations of the noise on distance measures σ_d. The reader can observe that the GC approach and the SDP with gradient refinement show exactly the same localization errors. This is, however, quite intuitive, since they reach the same optimum of the objective function. On the other hand, Table 2 reports the mean CPU time observed in the experiments. The SDP approach requires, in terms of CPU time, an order of magnitude more computational effort than the global continuation approach. This issue becomes critical as the network size increases, making the techniques practically unusable, for networks with more that 200 nodes. We remark that this problem

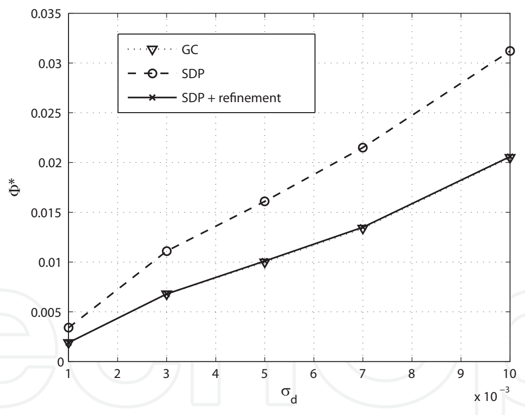


Fig. 11. Localization error of the configuration estimated with global continuation (dotted line with triangle), SDP (dashed line with circle) and SDP with gradient refinement (dotted line with cross).

Table 2. CPU time for global continuation (GC) approach, semidefinite programming and SDP with gradient refinement. Time is expressed in seconds.

σ_d	GC	SDP	SDP + GM
0.001	2.2942	20.1295	20.3490
0.003	2.3484	18.5254	18.9188
0.005	1.7184	16.5945	16.8349
0.007	1.7191	15.8923	16.1929
0.01	1.7360	15.8138	16.1093

has been addressed with some forms of distributed implementation of the SDP approach, see in (Biswas et al., 2006). Some discussion on distributed network localization is reported in the following section.

7. A Note on Distributed Range Localization

The techniques mentioned above are centralized since they require all the data needed for problem solution (i.e. distance measurements and anchor positions) to be available to a central units which perform network localization and then communicates to each node the estimated position. This may of course be highly undesirable due to intensive communication load over the central units and the agents. Moreover, since all the computation is performed by a single unit, for large networks the computational effort can be just too intensive. Also, the system is fragile, since failure in the central elaboration unit or in communication would compromise the functioning of the whole network. According to these considerations, *distributed* approaches are desirable for solving network localization. In a distributed setup each node communicates only with its *neighbors*, and performs local computations in order to obtain an estimate of its own position. As a consequence, the communication burden is equally spread among the network, the computation is decentralized and entrusted to each agent, improving both efficiency and robustness of the estimation process. Literature on decentralized network localization includes the application of distributed weighted-multidimensional scaling (Costa et al., 2006), and the use of barycentric coordinates for localizing the nodes under the hypothesis that non-anchor nodes lie in the convex hull of anchors (Khan et al., 2009). An extension of the SDP framework to distributed network localization can be found in (Biswas et al., 2006), whereas contributions in the anchor-free setup include (Xunxue et al., 2008). We conclude the chapter with a brief outline of a distributed extension of the gradient method presented in Section 4.1. We first notice that the gradient information which is needed by the node for an update step requires local-only information. Each node, in fact, can compute the local gradient as:

$$\nabla_i f(p) = \sum_{j \in \mathcal{N}_i} (p_i - p_j)^\top g_{ij}(p), \tag{25}$$

where $\nabla_i f(p)$ denote the i-th 1×2 block in the gradient $\nabla f(p)$ in (11) and $\mathcal{N}_i$ are the neighbors of the node i. It follows that the portion of gradient $\nabla_i f(p)$ can be computed individually by node i by simply querying the neighbors for their current estimated positions. For iterating the gradient method each node also needs the stepsize α_τ, which depends on some global information. The expression of the stepsize (5), however, is particularly suitable for decentralized computation, as we can notice by rewriting (5) in the following form:

$$\alpha_\tau = \frac{\sum_{i=1}^n \|p_i^{(\tau)} - p_i^{(\tau-1)}\|^2}{\sum_{i=1}^n (p_i^{(\tau)} - p_i^{(\tau-1)})^\top (\nabla_i f(p^{(\tau)}) - \nabla_i f(p^{(\tau-1)}))}, \tag{26}$$

It is easy to observe that each summand that composes the sum at the denominator and the numerator of α_τ is a local quantity available at node i. Hence a distributed averaging method, like the one proposed in (Xiao et al., 2006), allows each node to retrieve the quantities $\frac{1}{n} \sum_{i=1}^n \|p_i^{(\tau)} - p_i^{(\tau-1)}\|^2$ and $\frac{1}{n} \sum_{i=1}^n (p_i^{(\tau)} - p_i^{(\tau-1)})^\top (\nabla_i f(p^{(\tau)}) - \nabla_i f(p^{(\tau-1)}))$. By simply dividing these quantities each node can obtain the stepsize α_τ and can locally update its estimated position according to the distributed gradient rule:

$$p_i^{(\tau+1)} = p_i^{(\tau)} - \alpha_\tau \nabla_i f(p^{(\tau)}). \tag{27}$$

Similar considerations can be drawn about the Gauss-Newton approach. On the other hand it can be difficult to derive a distributed implementation of the global continuation and trust region approaches, limiting their effectiveness in solving the network localization problem.

8. Conclusion

In this chapter we review several centralized techniques for solving the network localization problem from range measurements. We first introduce the problem of information fusion aimed at the estimation of node position in a networked system, and we focus on the case in which nodes can take pairwise distance measurements. The problem setup is naturally modeled using graph formalism and network localization is expressed as an optimization problem. Suitable optimization methods are then applied for finding a minimum of the cost function which, under suitable conditions, corresponds to the actual network configuration. In the chapter we analyze five numerical techniques for solving the network localization problem under range-only measurements, namely a gradient method, an Gauss-Newton algorithm, a Trust-Region method, a Global Continuation approach and a technique based on semidefinite programming. The methods are described in details and compared, in terms of computational efficiency and convergence properties. Several tests and examples further define possible applications of the presented models, allowing the reader to approach the problem of position estimation in networked system paying attention to both theoretical and practical aspects. The

first three techniques (GM, GN and TR) are local in the sense that the optimization techniques are able to attain the global optimum of the objective function only when some initial guess on node configuration is available and this guess is sufficiently close to actual node positions. The convergence properties of these techniques are tested through extensive simulations. The gradient method can be implemented easily and requires only first order information. In this context we recall a simple and effective procedure for computing the stepsize, called Barzilai-Borwein stepsize. The Gauss-Newton approach, although being the fastest and most efficient method, is prone to convergence to local minima and it is therefore useful only when good a-priori knowledge of the node position is available. The trust-region method has better convergence properties with respect to the previous techniques, providing a good compromise between numerical efficiency and convergence. We also present two global approaches, a global continuation approach and a localization technique based on semidefinite programming (SDP). Global continuation, although computationally intensive, shows convergence to the global optimum regardless the initial guess on node configuration. Moreover it allows to compute accurate position estimates also in presence of noise. Finally the SDP approach is able to retrieve the exact node position in the case of noiseless distance measurements, by relaxing the original problem formulation. In the practical case of noisy measure, the approach tends to be inaccurate, and the localization error heavily depends on the number of anchor nodes and on their placement. In order to improve the localization accuracy we also discuss the possibility of adding a local refinement to the SDP estimate, evaluating this solution in terms of precision and computational effort.
We conclude the chapter by discussing how decentralized implementations of the network localization algorithms can be derived, and reviewing the state-of-the-art on distributed range-based position estimation.

9. References

Akyildiz, I., Su, W., Sankarasubramniam, Y. & Cayirci, E. (2002). A survey on sensor networks, *IEEE Communication Magazine* **40**(8): 102–114.

Barooah, P. & Hespanha, J. (2007). Estimation on graphs from relative measurements, *IEEE Control Systems Magazine* **27**(4): 57–74.

Barzilai, J. & Borwein, J. (1988). Two-point step size gradient methods, *IMA J. Numer. Anal.* **8**: 141–148.

Biswas, P., Lian, T., Wang, T. & Ye, Y. (2006). Semidefinite programming based algorithms for sensor network localization, *ACM Transactions on Sensor Networks (TOSN)* **2**(2): 220.

Biswas, P. & Ye, Y. (2004). Semidefinite programming for ad hoc wireless sensor network localization, *Proceedings of the Third International Symposium on Information Processing in Sensor Networks (IPSN)*, pp. 2673–2684.

Boyd, S., V. L. (2004). *Convex optimization*, Cambridge University Press.

Costa, J., Patwari, N. & Hero, A. (2006). Distributed weighted-multidimensional scaling for node localization in sensor networks, *ACM Transactions on Sensor Networks* **2**(1): 39–64.

Doherty, L., Pister, K. & El Ghaoui, L. (2001). Convex position estimation in wireless sensor networks, *IEEE INFOCOM*, Vol. 3, pp. 1655–1663.

Eren, T., Goldenberg, D., Whiteley, W., Yang, Y., Morse, A., Anderson, B. & Belhumeur, P. (2004). Rigidity, computation, and randomization in network localization, *IEEE INFOCOM*, Vol. 4, pp. 2673–2684.

Foy, W. (1976). Position-location solutions by Taylor-series estimation, *IEEE Transaction on Aerospace and Electronic Systems AES-12 (2)*, pp. 187–194.

Hendrickson, B. (1995). The molecule problem: Exploiting structure in global optimization, *SIAM Journal on Optimization* **5**(4): 835–857.

Howard, A., Mataric, M. & Sukhatme, G. (2001). Relaxation on a mesh: a formalism for generalized localization, *EEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'01)*.

Kannan, A., Mao, G. & Vucetic, B. (2006). Simulated annealing based wireless sensor network localization, *Journal of Computers* **1**(2): 15–22.

Khan, U., Kar, S. & Moura, J. (2009). Distributed sensor localization in random environments using minimal number of anchor nodes, *IEEE Transactions on Signal Processing* **57**: 2000–2016.

Mao, G., Fidan, B. & Anderson, B. (2007). Wireless sensor network localization techniques, *Computer Networks* **51**(10): 2529–2553.

Martinez, S. & Bullo, F. (2006). Optimal sensor placement and motion coordination for target tracking, *Automatica* **42**(4): 661–668.

Moore, D., Leonard, J., Rus, D. & Teller, S. (2004). Robust distributed network localization with noisy range measurements, *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 50–61.

Moré, J. & Wu, Z. (1997). Global continuation for distance geometry problems, *SIAM Journal on Optimization* **7**(3): 814–836.

More, J.J., S. D. (1983). Computing a trust region step, *SIAM Journal on Scientific and Statistical Computing* **4**: 553–57.

Niculescu, D. & Nath, B. (2001). Ad hoc positioning system (aps), *in Proceedings of IEEE GLOBECOM '01*, pp. 2926–2931.

Nie, J. (2009). Sum of squares method for sensor network localization, *Computational Optimization and Applications* **43**(2): 151–179.

Nocedal, J. & Wright, S. (2006). *Numerical Optimization*, Springer.

Priyantha, N., Balakrishnan, H., Demaine, E. & Teller, S. (2003). Anchor-free distributed localization in sensor networks, *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 340–341.

Savarese, C., Rabaey, J. & Langendoen, K. (2002). Robust positioning algorithms for distributed ad-hoc wireless sensor networks, *USENIX Annual Technical Conference*, pp. 317–327.

Saxe, J. (1979). Embeddability of weighted graphs in *k*-space is strongly NP-hard, *Proceedings of the 17th Allerton Conference in Communications, Control and Computing*, pp. 480–489.

Siciliano, B. & Khatib, O. (2008). *Springer Handbook of Robotics*, Springer-Verlag.

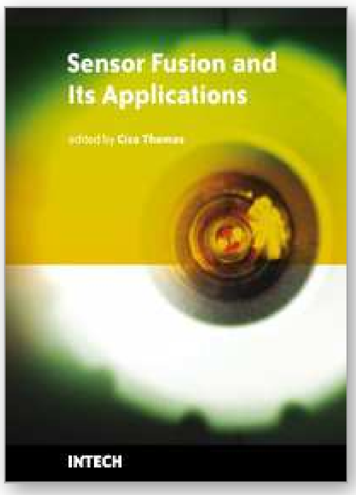
So, A. & Ye, Y. (2007). Theory of semidefinite programming for sensor network localization, *Mathematical Programming* **109**(2): 367–384.

Stanfield, R. (1947). Statistical theory of DF finding, *Journal of IEE* **94**(5): 762–770.

Tseng, P. (2007). Second-order cone programming relaxation of sensor network localization, *SIAM Journal on Optimization* **18**(1): 156–185.

Xiao, L., Boyd, S. & Lall, S. (2006). Distributed average consensus with time-varying Metropolis weights, *Unpublished manuscript* . `http://www.stanford.edu/~boyd/papers/avg_metropolis.html`.

Xunxue, C., Zhiguan, S. & Jianjun, L. (2008). Distributed localization for anchor-free sensor networks, *Journal of Systems Engineering and Electronics* **19**(3): 405–418.



Sensor Fusion and its Applications
Edited by Ciza Thomas

ISBN 978-953-307-101-5
Hard cover, 488 pages
Publisher Sciyo
Published online 16, August, 2010
Published in print edition August, 2010