Solving Ambiguities in MDS Relative Localization

Carmelo Di Franco, Alessandra Melani, Mauro Marinoni
Scuola Superiore Sant'Anna, Pisa, Italy
E-mail: {c.difranco, al.melani, m.marinoni}@sssup.it

***Abstract*—Monitoring teams of mobile nodes is becoming crucial in a growing number of activities. Where it is not possible to use fixed references or external measurements, one of the possible solutions involves deriving relative positions from local communication. Well-known techniques such as trilateration and multilateration exist to locate a single node although such methods are not designed to locate entire teams. The technique of Multidimensional Scaling (MDS), however, allow us to find the relative coordinates of entire teams starting from the knowledge of the inter-node distances. However, like every relative-localization technique, it suffers from geometrical ambiguities including rotation, translation, and flip. In this work, we address such ambiguities by exploiting the node velocities to correlate the relative maps at two consecutive instants. In particular, we introduce a new version of MDS, called enhanced Multidimensional Scaling (eMDS), which is able to handle these types of ambiguities. The effectiveness of our localization technique is then validated by a set of simulation experiments and our results are compared against existing approaches.**

I. Introduction

Teams of mobile robots are used to carry out a variety of tasks, such as surveillance, monitoring, exploration, search and rescue, cleaning of hazardous areas, and transportation. A representative example is given by localization systems that are used to track and support human workers operating in dangerous areas [1].

To be able to coordinate effectively between members of the team, it is necessary to exchange information about the environment so that each node can locate the entire team and perform the assigned activities properly. In some situations, it may be possible figure out the absolute positions by exploiting an infrastructure that already exists (e.g. Global Positioning System (GPS)) or one that has potential high costs but that can be designed from scratch. Moreover, this option may not be viable in emergency situations. Using local communication to calculate the inter-node distances and then creating a relative map of the team is a common solution when absolute localization is not feasible. There are other well-known techniques including trilateration and multilateration , but these only permit us to locate single nodes. In recent years, much attention has been paid to the Multidimensional Scaling (MDS) technique [2], which is widely used in various fields for scientific visualization and data mining [3] and also used in the field of robot localization to find relative maps of the nodes. MDS is a method that represents measurements of similarity (or dissimilarity) among pairs of objects as distances between points in an N-dimensional space. However, like every relative localization technique, it suffers from geometrical ambiguities that make it difficult to correlate the coordinates of the nodes that are calculated at successive time instants. More specifically, even with no measurement errors, the original topology of the network can only be estimated up to an isometric transformation (i.e., rotation, translation, and flip).

978-1-4673-7509-2/15/$31.00

A. Contributions and summary

This work proposes a new implementation of MDS, namely enhanced Multidimensional Scaling (eMDS), which correlates the node-distances information at two consecutive time instants with their velocities to solve all the geometric ambiguities caused by inter-node relative distances. On one hand, the proposed approach inherits from MDS the advantage of estimating the coordinates of all nodes jointly, even for large teams of nodes. On the other hand, it overcomes MDS limitations by correlating the relative maps at consecutive instants. Furthermore, our eMDS technique allows a more advanced coordination, as each node can not only locate the team, but can also exploit the knowledge on the nodes velocities to predict future movements.

The rest of the paper is divided into five sections. Section II analyzes the state of the art, describing the existing solutions to the problem. Section III reviews the MDS technique, focusing on the ambiguities introduced by the original method. Section IV describes the localization system we consider, and the eMDS technique is proposed and formalized in more detail. In Section V, we carry out a quantitative evaluation of the algorithm in presence of noisy measurements. Finally, Section VI draws conclusions and outlines future work.

II. Related Work

Relative localization is an important issue in many fields. Several wireless communication techniques have been utilized to address the problem of locating nodes within a relative coordinate system [4]. Many approaches present in literature are based on the Simultaneous Localization and Mapping (SLAM) [5] that constructs the map and the nodes positions at the same time. However, these solutions are composed of numerous steps and typically require multiple sensors and a significant amount of computation to improve the quality of localization. The other main family includes the algorithms that acquire the positions of the nodes starting from inter-node distances. An important advantage of the latter solutions is the possibility it provides to obtain information on distances from the communication infrastructure, thus reducing the need of auxiliary sensors. The rest of the paper will be focused on this last family due to the exposed advantages that make them suitable for small and low-cost nodes.

Trilateration or multilateration are often used in distance-based localization. Nevertheless, relative localization suffers from ambiguities in terms of translation, rotation, and flip. Whenever trilateration or multilateration are used, flip ambiguities also arise during the map construction process, since the positions of the nodes are estimated one at a time. Kannan et al. [6] have provided a formal geometric analysis of flip ambiguity problems, which are possible sources of computational corruption in trilateration-based algorithms. Several authors proposed distributed techniques that can solve the problem in the case of large (static) wireless networks [7], [8], [9].

However, in several multi-agent applications of mobile robots, a node is required to locate the position of multiple team members simultaneously. This can be achieved efficiently by the MDS [10], a general approach for exploiting similarities in data to assign a location to each item in an N-dimensional space. In the context of robot localization, MDS is used to acquire the coordinates of a group of nodes. This is achieved by minimizing the mismatch between a set of pairwise-estimated inter-node distances and the distances corresponding to the unknown coordinates. Remarkably, unlike trilateration and multilateration, MDS-based techniques do not suffer from the problem of flip ambiguities during the map construction process, since they are able to find all the coordinates consecutively.

Many variants of MDS have been introduced in the literature (see [10] and references therein). Among these variants, classical MDS constructs the relative map of the nodes using the eigenvectors corresponding to the N largest eigenvalues of the dissimilarity matrix, while non-classical MDS resorts to an iterative method that minimizes a stress function. In the latter category, weighting schemes have also been proposed to increase the robustness of the minimization procedure in case of missing information [11]. Efatmaneshnik et al. [12] proposed a modified version of MDS targeting Vehicular Adhoc NETworks (VANETs) where nodes can be equipped with GPS, which provides a priori information to be fused with short-range measurements.

All of the techniques mentioned above either require the use of a fixed infrastructure (e.g., GPS or anchors), or suffer the problem of correlating the maps produced consecutively. In the literature, the problem of correlating the information of two consecutive outputs of MDS has been widely studied: Ambrosi and Hansohm [13] described a dynamic Multidimensional Scaling (dMDS) method that makes use of a super-dissimilarity matrix that includes information regarding T instants. However, in the context of localization, this technique cannot be applied at all due to cross-time inter-node distance information not being available. A different approach can be taken with dMDS, by simply carrying out an MDS analysis for each time period separately, and then matching the resulting configurations using a Procrustes analysis [10]. However, both methods only aim to plot the data together without any strict physical correlations. Cabero et al. [14] used dMDS for indoor people tracking. However, they also make use of a large number of anchors which means that the problems of flip, rotation, and translation become irrelevant. Other works tried to improve MDS localization by exploiting inertial measurements [15], but only mitigating the inconsistencies induced by the MDS algorithm.

To our knowledge, only two works in the literature address the problem of mobile robot localization without anchors. Oliveira and Almeida [16] in particular proposed a technique that gives confidence values to position estimates obtained by MDS at successive instants. Instead, Beck and Baxley [17] proposed an anchor-free node tracking technique that uses dMDS and odometry. They successfully locate the nodes at several instants, but the drawback of this approach is that the computational complexity proportionally grows with the number of nodes and instants, making it unsuitable for online localization. This work proposes an improved implementation of MDS, called eMDS, where the nodes velocities are used to solve the ambiguities generated by two consecutive MDS outputs. Unlike in [17], the computational complexity of the algorithm only depends on the number of nodes, and produces relative coordinates online. Our technique can not only provide the coordinates of the nodes, but also their likely future movements, providing more information to any coordination algorithm.

III. The MDS algorithm and its ambiguity problem

This section reviews the MDS algorithm, with close attention paid to the issues related to its use in the context of relative localization.

The algorithm takes pairwise (dis)similarities (e.g., Euclidean distances) as input data, grouped in a distance (or dissimilarity) matrix D and returns a set of estimated coordinates $\tilde{X} = [\vec{x}_1, ..., \vec{x}_n]$, as shown in Figure 1. More

Fig. 1. The Multidimensional Scaling algorithm.

specifically, given n nodes in an N-dimensional space and the estimated pairwise distances d_{ij}, MDS estimates the nodes coordinates $\vec{x}_i$ for all nodes minimizing the mismatch between the estimated distances $\widehat{d}_{ij}$ and the distances $\|\vec{x}_i - \vec{x}_j\|$. The mismatch S_{MDS} is called *stress function*:

$$S_{MDS}(\vec{x}_1, ..., \vec{x}_n) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} [\|\vec{x}_i - \vec{x}_j\| - \widehat{d}_{ij}]^2. \tag{1}$$

However, since only pairs of range measurements are used, the relative map $\tilde{X}$ produced by MDS represents the correct positions of the nodes up to translation, rotation, and flip. Figures 2 and 3 illustrate an example that shows the real positions of the nodes in a bidimensional space (a) and the corresponding MDS output (b) at two successive instants t and t' with no error in the distance measurements. In both cases, as expected, MDS estimates the correct relative coordinates up to translation, rotation, and flip, since there are no errors. However, since at every MDS instance the resulting map is affected by different ambiguities, it is difficult to i) apply any filtering to the MDS output if nodes are moving, since this would require a characterization of the noise and the knowledge of the node dynamic; ii) understand where the nodes are moving or heading; iii) apply information on the absolute orientation (given by the compass) directly on the MDS output. In [16], confidence values are obtained between time instants using distance information only, while in [17] a solution is proposed that exploits dMDS to correlate information gathered at consecutive instants. However, both approaches are only able to solve the first of the requirements mentioned above.

In this work, we propose a modified version of MDS that not only provides a correct correlation between time instants, but also inform us of nodes' direction of travel and a prediction of their likely future movements. Since we only need the distance information at the previous and current instants, the algorithm is well-suited for online localization with fast dynamics.

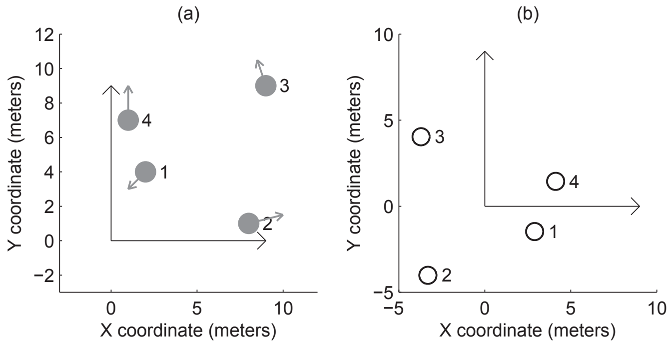

Fig. 2. (a) Real positions of four nodes at instant t, with outgoing arrows representing the speed vectors of the nodes. (b) MDS output at time instant t. The topology of the network is preserved up to an isometric transoformation.

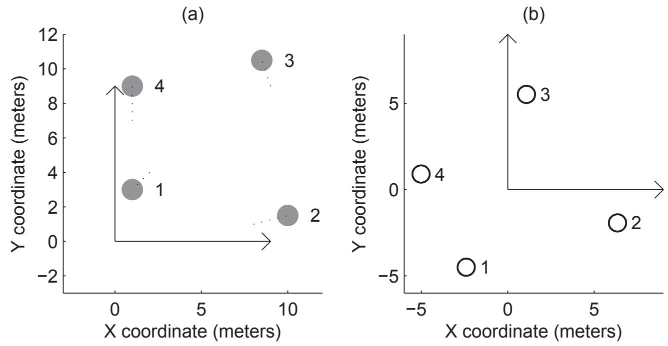

Fig. 3. (a) Real positions of the nodes at the successive instant t', with incoming dashed lines representing the traveled distance. (b) Corresponding MDS output at instant t'. The topology of the network is completely unrelated to the previous MDS output.

A. MDS with limited range and communication errors

The MDS algorithm requires a symmetric distance Matrix D as input. However, since distances estimated by RF signals are subject to errors, the measured distance $\widehat{d_{ij}}$ may be different from $\widehat{d_{ji}}$, leading to a non-symmetrical matrix. In this case, D must be converted into a symmetrical form. Possible solutions which are commonly adopted in the literature [18] are to calculate the mean value of $\widehat{d_{ij}}$ and $\widehat{d_{ji}}$, or to simply select one of the two as a unique value. If instead a reliability weight is associated to each distance information (e.g. an indicator of the link quality), another possible solution could be to select that which has the highest weight. Moreover, due to limited range, it may happen that not all pairwise distances are known. However, a fully connected matrix is required as an input to MDS. Several techniques have been proposed in the literature to overcome this issue. For instance, it is possible to exploit the network topology to obtain the missing distances [19], or use a modified version of MDS for partially connected matrices [20].

In this work, we over-approximate the missing distances between pairs of nodes i and j by adding the distances along the shortest path composed of intermediate nodes, as previously done in [18]. The shortest path can be calculated by Dijkstra or Floyd-Warshall algorithm [19], and then MDS can be regularly applied. This technique provides a valid approximation of the missing values, since the distance between two disconnected nodes is shorter or equal than the distance between the sum of the nodes that indirectly connect them (e.g. $d_{13} \leq d_{12} + d_{23}$). Although pessimistic, this approximation is acceptable in our localization system, since, for the purpose of team localization, an imprecise estimation of the positions of distant nodes can be tolerated from time to time. Additionally, some packets may be lost during local communication. In particular, if $\widehat{d_{ij}}$ is not available due to packet loss, such distance information can be replaced by $\widehat{d_{ji}}$, if available. An alternative, also implemented in this work, is to predict unavailable measurements with some filtering techniques. This filter may also allow reducing fluctuations due to inaccuracy of sensor measurements (e.g. RSSI). To address these problems, bayesian filters [21] are widely used in literature. Amongst the most used include particle filters and Kalman filters. Section IV-C presents the Kalman Filter used in this work together with its implementation details.

IV. Proposed approach

In this section the system description is presented along with the complete formulation of our improved eMDS technique.

A. System description

In this work, we consider a system composed of n mobile nodes in a two-dimensional space that communicate through a radio channel and form fully connected network. The real coordinates, real velocities, and real accelerations of the nodes at every time instant t are denoted by X,V, and A:

$$X(t) = \begin{bmatrix} \vec{x}_{1,t} \\ \vec{x}_{2,t} \\ \vdots \\ \vec{x}_{n,t} \end{bmatrix}, V(t) = \begin{bmatrix} \vec{v}_{1,t} \\ \vec{v}_{2,t} \\ \vdots \\ \vec{v}_{n,t} \end{bmatrix}, A(t) = \begin{bmatrix} \vec{a}_{1,t} \\ \vec{a}_{2,t} \\ \vdots \\ \vec{a}_{n,t} \end{bmatrix}. \quad (2)$$

Each node is equipped with a radio system that is allowed to estimate the inter-node distances (e.g., Received Signal Strength Indicator (RSSI), Time of Flight (ToF), Ultra-WideBand (UWB)) and that can estimate its velocity with dedicated sensors (e.g., Inertial Measurement Unit (IMU), compass, optical flow smart camera, odometry) that are selected according to the specific needs of the application. Every node shares its information with the entire team through a TDMA-like communication protocol, and internally estimates the distance traveled since the previous transmission or sends enough information to reconstruct it in the receiving node. The selection of the transmitted data and the estimation approach both depends on the nodes' dynamic and on the specific available sensors. In our scenario we assume a slow change in the speed between two consecutive transmissions, thus each node obtains all the new inter-node distances and the estimated velocities at every TDMA round, storing this information in a distance matrix $\widehat{D}$ and a velocity vector $\widehat{V}$:

$$\widehat{D}(t) = \begin{bmatrix} \widehat{d}_{1,1,t} & \widehat{d}_{1,2,t} & \dots & \widehat{d}_{1,n,t} \\ \widehat{d}_{2,1,t} & \widehat{d}_{2,2,t} & \dots & \widehat{d}_{2,n,t} \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{d}_{n,1,t} & \widehat{d}_{n,2,t} & \dots & \widehat{d}_{n,n,t} \end{bmatrix}, \widehat{V}(t) = \begin{bmatrix} \widehat{\vec{v}}_{1,t} \\ \widehat{\vec{v}}_{2,t} \\ \vdots \\ \widehat{\vec{v}}_{n,t} \end{bmatrix}.$$

According to this discrete time model, the velocity of each node is assumed to be constant over each TDMA round.

Figure 4 clarifies the structure of the entire system: at time t, each node computes the relative map $\widehat{X}(t)$ using eMDS, which will be described in detail in Section IV-B, and the result is filtered with a Kalman Filter (KF). The output of the KF, denoted as $\tilde{X}(t)$, is then used as a feedback to reduce the ambiguities.

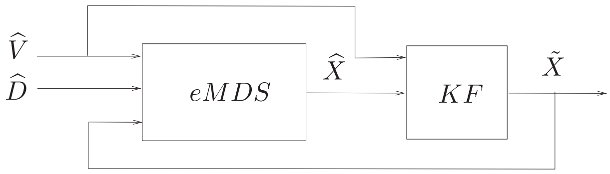

Fig. 4. Model of the proposed relative localization system.

B. The enhanced Multidimensional Scaling

A different version of the MDS, denoted as eMDS, is proposed to solve the problem of correlating two MDS outputs at consecutive instants. MDS, as shown before, performs a non-linear unconstrained optimization where n coordinates are evaluated starting from a distance matrix. We propose to reformulate such optimization as a non-linear constrained problem with $2n$ variables. The stress function considers the mismatch between estimated and real distances at two consecutive instants, while the constraints are used to correlate real positions relying on the knowledge of the node velocities.

The formulation of eMDS as optimization problem is then:

$$\begin{aligned} \min \quad & S(\vec{x}_{1,t-1}, \dots, \vec{x}_{n,t-1}, \vec{x}_{1,t}, \dots, \vec{x}_{n,t}) \\ subject\ to \quad & \begin{cases} \vec{x}_{1,t} = \vec{x}_{1,t-1} + \vec{v}_{1,t-1} \cdot \Delta t \\ \vec{x}_{2,t} = \vec{x}_{2,t-1} + \vec{v}_{2,t-1} \cdot \Delta t \\ \dots \\ \vec{x}_{n,t} = \vec{x}_{n,t-1} + \vec{v}_{n,t-1} \cdot \Delta t \end{cases}, \end{aligned} \tag{3}$$

where S is the stress function, defined for $t > 0$ as:

$$\begin{aligned} S = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} [(\|\vec{x}_{i,t} - \vec{x}_{j,t}\| - \widehat{d}_{ij,t})^2 + \\ + (\|\vec{x}_{i,t-1} - \vec{x}_{j,t-1}\| - \widehat{d}_{ij,t-1})^2]. \end{aligned} \tag{4}$$

The linear equality constraints in Equation (3) can be rewritten in matrix form as $AX = b$, where:

$$A = \begin{bmatrix} 1 & 0 & \dots & 0 & -1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & -1 \end{bmatrix},$$

$$X = \begin{bmatrix} \vec{x}_{1,t-1} \\ \vdots \\ \vec{x}_{n,t-1} \\ \vec{x}_{1,t} \\ \vdots \\ \vec{x}_{n,t} \end{bmatrix}, b = \begin{bmatrix} \vec{v}_{1,t-1}\Delta t \\ \vdots \\ \vec{v}_{n,t-1}\Delta t \\ \vec{v}_{1,t}\Delta t \\ \vdots \\ \vec{v}_{n,t}\Delta t \end{bmatrix}.$$

Then, since the speed estimation is affected by a non-zero error ϵ_v, it is necessary to reformulate the linear constraints as inequalities, i.e., $|AX - b| \leq \epsilon_v$. Hence, in the general case, the constraints in Equation (3) become

$$\begin{cases} AX \leq b + \epsilon_v \\ -AX \leq -b + \epsilon_v \end{cases}.$$

The stress function S is coercive (i.e., $S(X) \to +\infty$ as $||X|| \to +\infty$), but not convex. Hence it is guaranteed to have a global minimum, but may present multiple local minima. An accurate choice of the starting point for the minimization could lead to the discovery of the global minimum. To help avoid local minima, most implementations of MDS attempt multiple random initial configurations. In this work, instead of using random initial points, we exploit the positions and velocities estimated at the previous instant. In particular, the starting point X_0 used in the minimization is set as follows: the first n variables are set as equal to the vector of coordinates $\tilde{X}(t)$ obtained at the previous step, and the last n are set as $\tilde{X}(t) + V\Delta t$. With zero-noise in the measurements, the eMDS finds the global minimum in few algorithm iterations. After finding the global minimum for the first time, it will continue to find all the future global minima. However, there is a particular case where not all ambiguities can be solved, i.e., the case where the nodes move at the same constant speed and with the same orientation. When dMDS is used (e.g., in [17]), this configuration cannot be distinguished from the case where all the nodes are moving back and forth, since dMDS does not account for the node orientation but only for the distance traveled. Instead, with our approach the problem appears, unless the nodes start moving in this particular configuration. In the latter case, the algorithm can solve ambiguities up to a flip (with respect to the speed direction), which cannot be disambiguated until one of the nodes changes its direction.

C. Filtering the eMDS with a Kalman Filter

In the presence of noisy distance estimates, the eMDS minimization procedure may find some local minima corresponding to a flipped configuration with respect to the real one. This effect is not caused by the algorithm itself, as in the case of the original MDS. Rather, it is the inaccuracy of the estimated distances that may lead to an output topology that are compatible with the erroneous estimated distances but not with those that are accurate. These flipped configurations can easily be filtered by implementing a filter for each node. The state equation of each Kalman Filter, used in this work, is defined as follows:

$$P^s = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} P^{s-1} + \begin{bmatrix} \frac{\Delta t^2}{2} & 1 \\ 0 & \Delta t \end{bmatrix} \psi^s; \tag{5}$$

$$P = [\vec{x} \quad \vec{v}\,], \tag{6}$$

whereas the state vector P is defined in Equation (6) and ψ^s is the Gaussian noise of the state at instant s. The measurement equation is defined as:

$$Y^s = P^s + \nu^s, \tag{7}$$

where ν^s is the Gaussian noise of the measurement.

V. SIMULATION RESULTS

This section presents a simulation study carried out to test and validate the localization system proposed in this paper. In the following experiments, we aim to evaluate the impact of the measurement noise on the performance of eMDS and its effectiveness in solving geometrical ambiguities. We are not interested in studying the robustness of the approach with respect to the underlying communication protocol (e.g., communication delay, packet losses, hidden nodes), which can be addressed using techniques similar to those presented in Section III-A. Thus, we could assume without loss of generality that the simulations occur under the hypothesis of ideal communication, i.e., the network is fully connected and there are no packet losses.

A. Simulation Setup

Our methodology has been tested through a set of simulation experiments conducted in MATLAB®. In order to perform a simulation of our proposed approach on realistic data, we model our sensor measurements using real values estimated with specific technologies; in particular we consider as a reference platform a small low-cost wheel robot equipped with encoders and a compass that communicates using a 2.4GHz transceiver capable of performing ToF measurements. The range measurements used in our simulations are perturbed by a gaussian noise with zero mean and standard deviation $\sigma_d = 0.6$ m. Table I summarized the typical values of σ_d for different wireless technologies.

	UWB	ToF	RSSI
σ_d	0.2 m	0.6 m	2 m

TABLE I. TYPICAL VALUES FOR THE STANDARD DEVIATION σ_d OF RANGE MEASUREMENTS WITH DIFFERENT TECHNOLOGIES.

Furthermore, we have considered an error on the velocity measurements that is consistent with a generic encoder wheel, i.e., a few centimeters per meter. We have modeled the error of the speed estimates accordingly as a gaussian noise with zero mean and standard deviation $\sigma_v = 0.05$ m/s. The orientation of the velocity vector is given by a compass with an error below one degree, hence in this setup the orientation has a negligible error. The initial positions of the nodes are randomly generated in a 15 m by 15 m arena. At each time instant, the module of the speed is randomly varied, and the orientation of each node is increased by an angle selected in the interval $[-\pi/2, \pi/2]$.

B. Experimental Results

The first simulation experiment aims to demonstrate how the error of the estimate varies depending on σ_d. Figure 5 reports the obtained results and shows that our method achieves a highly decent performance even when there are significant errors. We calculate the error of the estimate as the difference between the estimated positions and the real ones. In the box plot, the central mark indicates the median, the edges of the box indicate the 25^{th} and 75^{th} percentiles, the whiskers

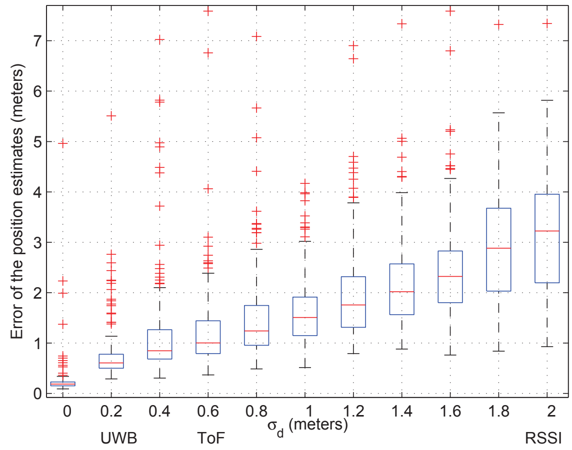

Fig. 5. Error of the position estimates as a function of σ_d.

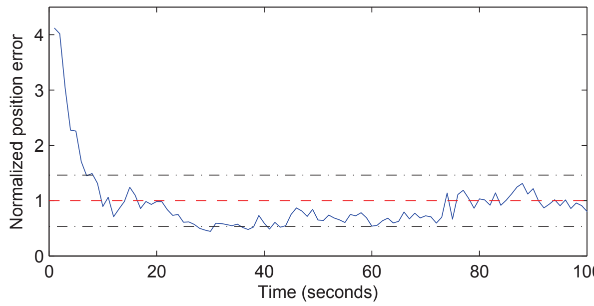

Fig. 6. Convergence of eMDS. The dash-dot lines represent the standard deviation, while the dashed line represents the mean error.

extend to the most extreme data points that are not considered outliers, and outliers are plotted individually as red marks. In this experiment, we have simulated a team of 6 nodes, and the parameter σ_d has been varied from 0 to 2 m at steps of 0.2. For each step, we have run eMDS for 200 time instants in order to characterize the behavior of the algorithm in detail. The image clearly demonstrates that the mean error increases almost linearly depending on σ_d, and the average values are always below 3.5 m even for high values of σ_d. Most of the outliers in Figure 5 are due to an initial settling time needed to reach the mean error, during which eMDS may be stuck in some local minima. Thanks to the accurate selection of the starting point for the minimization procedure which was previously discussed in Section IV-B, the algorithm then converges to its steady state. In order to evaluate the convergence time, we normalized all the data collected in the previous experiment with respect to their mean error. Figure 6 shows that in this case the algorithm takes 9 seconds to converge within the tolerance region which is fixed to the standard deviation. After reaching the steady state, subsequent local minima are filtered out by the KF.

We have also observed the confidence of the measurements, calculating the percentage of estimated positions contained inside different scales of the covariance ellipse. In Figure 7 we report a snapshot of an experiment where all the nodes are inside the 3-standard deviation ellipse, and their majority lies inside the 1-standard deviation ellipse, revealing the great accuracy of the predicted positions. In this particular

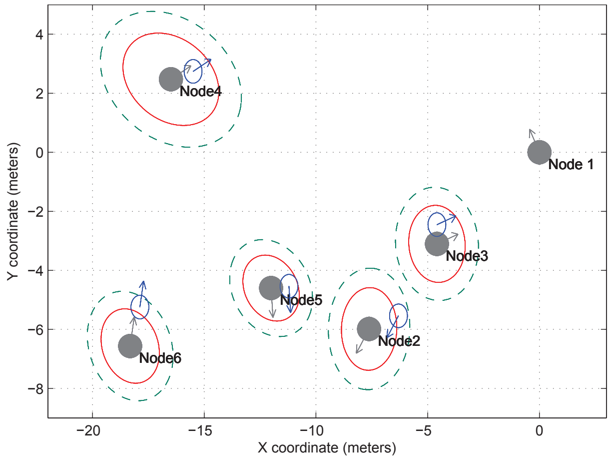

Fig. 7. Snapshot of the simulation field at a particular time instant t. The filled circles represent the nodes at real positions and the arrow represents the real velocity. Similarly, the empty circles with the corresponding arrows are the position and velocity estimates. The filled nodes are surrounded by the 1-standard deviation ellipse (full-line) and the 3-standard deviation ellipse (dashed-line).

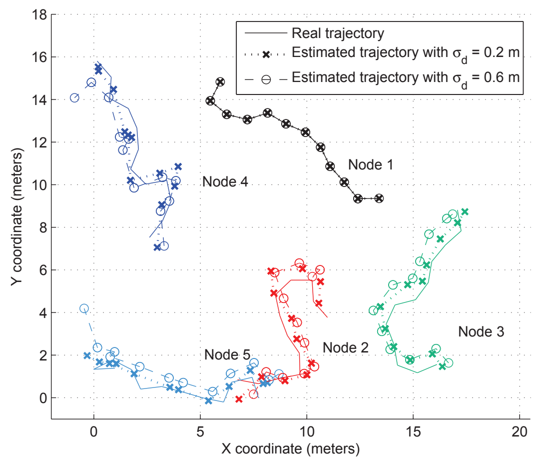

Fig. 8. Snapshot of the simulation field for 10 instants, with five nodes, $\sigma_v = 0.05$ m/s, $\Delta t = 1$ s and two different values of σ_d. The full lines represent the path of the nodes, the dotted lines with cross markers indicate the position estimates with $\sigma_d = 0.2$ m and the dashed lines with circle markers represent the position estimates with $\sigma_d = 0.6$ m. To make a comparison, all the trajectories of the nodes are translated with respect to the real positions of Node 1.

experiment, we have set $n = 6$, $\sigma_d = 0.6$ m, $\sigma_v = 0.05$ m/s, and $\Delta t = 1$ s. Another snapshot, reported in Figure 8, illustrates the accuracy of the node estimates with respect to the real paths followed by the nodes. Our proposed method allows achieving a correct correlation across different time instants, and nullifies all geometric ambiguities introduced by the original MDS algorithm.

We have also conducted an extensive experiment to evaluate the error on the position estimates depending on the number of nodes n and time step Δt. For this experiment, we have set $\sigma_d = 2$ m and $\sigma_v = 0.05$ m/s, and the number of nodes has been varied between 3 and 11. Again, the simulation has been performed for 200 instants for each value of n. The results are reported in Figure 9, where we observe that, the node movement is comparable with σ_d for low values of Δt (i.e., in the range 0-1 seconds),meaning that it is not possible to distinguish the node movement from the measurement noise.

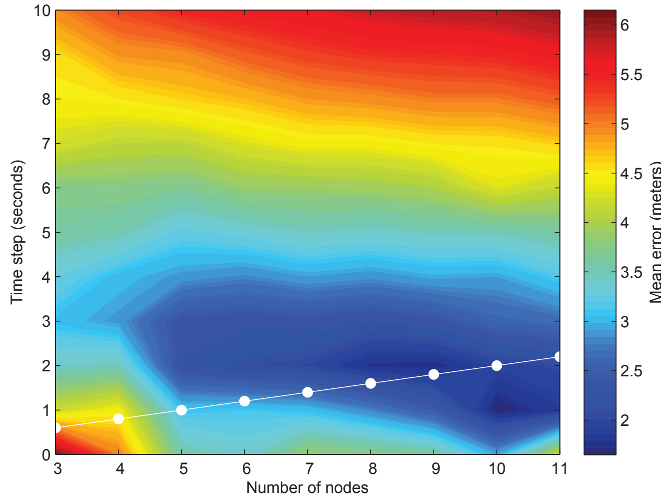

Fig. 9. Error of the position estimates as a function of the number of nodes n and the time step Δt. The dotted white line represents a single experiment, where $\Delta t = 0.2n$ s.

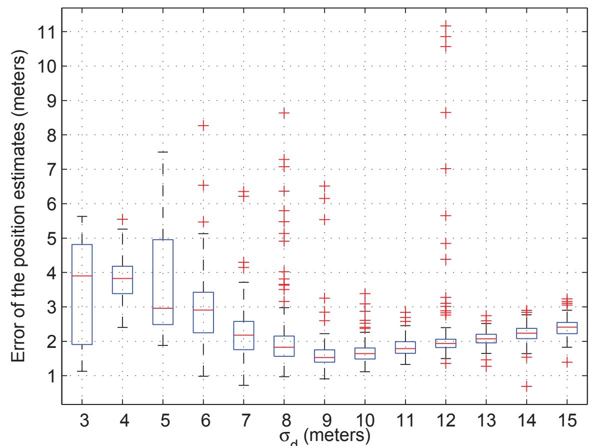

Fig. 10. Error of the position estimates as a function of the number of nodes n, with $\Delta t = 0.2n$ s.

Then, the increase of the team size has a beneficial effect on the mean error, since the algorithm can exploit more information to construct the relative map. However, this positive trend continues only up to a certain value of Δt (between 1 and 3 seconds in our case). In fact, from that point on, the communication latency is the prevailing effect, progressively leading to a slight error increase. From this experiment we can conclude that the increase of the team size and the time step are two conflicting effects balancing each other, and the best performance can be obtained as a trade-off between these two quantities. The dotted white line in Figure 9 represents a particular experiment, the results of which are illustrated in Figure 10. In this case, Δt is computed as $0.2n$ seconds, resembling the TDMA communication round. As expected, as the number of nodes increases(and, accordingly, Δt), the error starts decreasing. After $n = 9$, the communication delay leads to an error increase, due to the combination of the above-mentioned effects.

Finally, we compared our results with existing solutions. Oliveira and Almeida [16] reach a mean error μ in the position estimates equal to 1.3 m starting from a standard deviation in the measurement noise $\sigma_d = 0.6$ m and 99% of the errors are under 5 m. With the same measurement noise, our algorithm achieves the same mean error but the 99% of the errors are below 2.65 m. Beck and Baxley [17] solve the ambiguity problem with the dMDS technique, which takes a super-distance matrix of size $n \times k$, where n is the number of nodes and k is the number of time instants that we want to correlate, as input. Their algorithm has a number of variables equal to the size of the matrix, with an overall complexity of $O(k^2 n^2)$. Instead, our proposed technique correlates only two time instants, hence the number of variables in the optimization problem is $2n$. The overall complexity of eMDS is $O(n^2)$, since the stress function in Equation (4) is expressed as a double sum over n. The authors of [17] show that the position error tends to increase with k and to decrease with n. Compared to their work, the complexity of our approach only depends on the number of nodes, and the mean error is not affected by the time step, since only two time instants are considered at each execution. A direct comparison with their simulation results reveals that, with $\sigma_d = 0.1$ m and $\sigma_v = 0.01$ m/s, our mean error ($\mu = 0.427$ m) is equal to their best case (reached for $k = 2$), and always lower in all other cases. Moreover, a notable advantage of our method is that it is able to produce the relative coordinates online instead of calculating the entire paths offline.

VI. Conclusion and future work

This paper presented a novel approach for relative localization, called eMDS, which solves all the geometrical ambiguities introduced by the original MDS algorithm. In particular, MDS has been reformulated as a constrained nonlinear optimization problem to provide correct correlation between relative maps at consecutive instants. The error of the position estimates has been calculated as a function of the measurement error, and the proposed method achieves high quality performance in all of the tested configurations. In addition, our algorithm has been compared to current state-of-the-art techniques, and has been shown to outperform them, both in terms of complexity and error in the position estimates. As a notable aspect, the proposed technique overcomes some important limitations of current state-of-the-art techniques. Firstly, it is suitable for online team localization and does not require any fixed infrastructure, unlike most existing online approaches which require the use of anchors to localize the robot paths. Secondly, the coherence across time instants provides us with the opportunity to predict likely future movements, exploiting the knowledge on the node velocities. In the future, our work will be extended in multiple directions. First, we plan to test eMDS in a real environment with a team of mobile robots. Real experiments will validate our algorithm in terms of computational requirements, convergence time, and communication delays. Moreover, the Kalman filter is not the most suitable bayesian filter, due to its sensitivity with regards to noise statistics. A more efficient technique that we would like to explore is the Daum Huang filter, correlating it with the node dynamics. It would also be interesting to further extend eMDS to be applied in a tridimensional space. However, moving beyond bidimensional into tridimensional space will most likely require more information (e.g., the height of the nodes from the ground) and more constraints, in order to solve all the ambiguities that may arise. Finally, we plan to include acceleration to track nodes with fast dynamics, which cannot be modeled by assuming constant speed between consecutive position estimates.

Acknowledgments

The authors would like to thank Enrico Bini for his valuable comments and insightful discussions that inspired this work.

References

[1] A. Fink, H. Beikirch, M. Vo, and C. Schrder, "RSSI-based Indoor Positioning using Diversity and Inertial Navigation," in *Proceedings of the 5th IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2010.

[2] I. Borg and P. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, ser. Springer Series in Statistics. Springer, 2005. [Online]. Available: http://books.google.it/books?id=duTODldZzRcC

[3] Z.-X. Chen, H.-W. Wei, Q. Wan, S.-F. Ye, and W.-L. Yang, "A supplement to multidimensional scaling framework for mobile location: A unified view," *IEEE Transactions on Signal Processing*, vol. 57, no. 5, pp. 2030–2034, 2009.

[4] G. Mao, B. Fidan, and B. Anderson, "Wireless sensor network localization techniques," *Computer networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[5] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (slam): Part i the essential algorithms," *IEEE ROBOTICS AND AUTOMATION MAGAZINE*, vol. 2, p. 2006, 2006.

[6] A. Kannan, B. Fidan, G. Mao, and B. Anderson, "Analysis of flip ambiguities in distributed network localization," *Information, Decision and Control (IDC)*, pp. 193–198, 2007.

[7] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proceedings of the 2nd ACM International Conference on Embedded Networked Sensor Systems*, 2004, pp. 50–61.

[8] O.-H. Kwon, H.-J. Song, and S. Park, "Anchor-free localization through flip-error-resistant map stitching in wireless sensor network," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 11, pp. 1644–1657, 2010.

[9] P. Moravek, D. Komosny, M. Simek, and J. Muller, "Multilateration and flip ambiguity mitigation in ad-hoc networks," *Przegld Elektrotechniczny (Electrical Review)*, vol. 2012, no. 5b, pp. 222–229, 2012.

[10] T. F. Cox and M. A. Cox, *Multidimensional scaling*. CRC Press, 2010.

[11] J. A. Costa, N. Patwari, and A. O. Hero III, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 1, pp. 39–64, 2006.

[12] M. Efatmaneshnik, A. Balaei, N. Alam, and A. Dempster, "A modified multidimensional scaling with embedded particle filter algorithm for cooperative positioning of vehicular networks," in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE, 2009, pp. 7–12.

[13] K. Ambrosi and J. Hansohm, *Ein Dynamischer Ansatz zur Reprsentation von Objekten*, ser. Operations Research Proceedings. Springer Berlin Heidelberg, 1987, vol. 1986.

[14] J. M. Cabero, F. De la Torre, A. Sanchez, and I. Arizaga, "Indoor people tracking based on dynamic weighted multidimensional scaling," in *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, 2007, pp. 328–335.

[15] C. Di Franco, G. Franchino, and M. Marinoni, "Data fusion for relative localization of wireless mobile nodes," in *Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES)*. IEEE, 2014, pp. 58–65.

[16] L. Oliveira and L. Almeida, "Rf-based relative position estimation in mobile ad-hoc networks with confidence regions," in *RoboCup International Symposium*, 2014.

[17] B. Beck and R. Baxley, "Anchor free node tracking using ranges, odometry, and multidimensional scaling," in *Proceedings of the 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 2209–2213.

[18] L. Oliveira, H. Li, L. Almeida, and T. E. Abrudan, "Rssi-based relative localisation for mobile robots," *Ad Hoc Networks*, 2013.

[19] Y. Shang, W. Rumi, Y. Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 15, no. 11, pp. 961–974, 2004.

[20] A. Amar, Y. Wang, and G. Leus, "Extending the classical multidimensional scaling algorithm given partial pairwise distance measurements," *Signal Processing Letters, IEEE*, vol. 17, no. 5, pp. 473–476, 2010.

[21] J. V. Candy, *Bayesian signal processing: Classical, modern and particle filtering methods*. John Wiley & Sons, 2011, vol. 54.