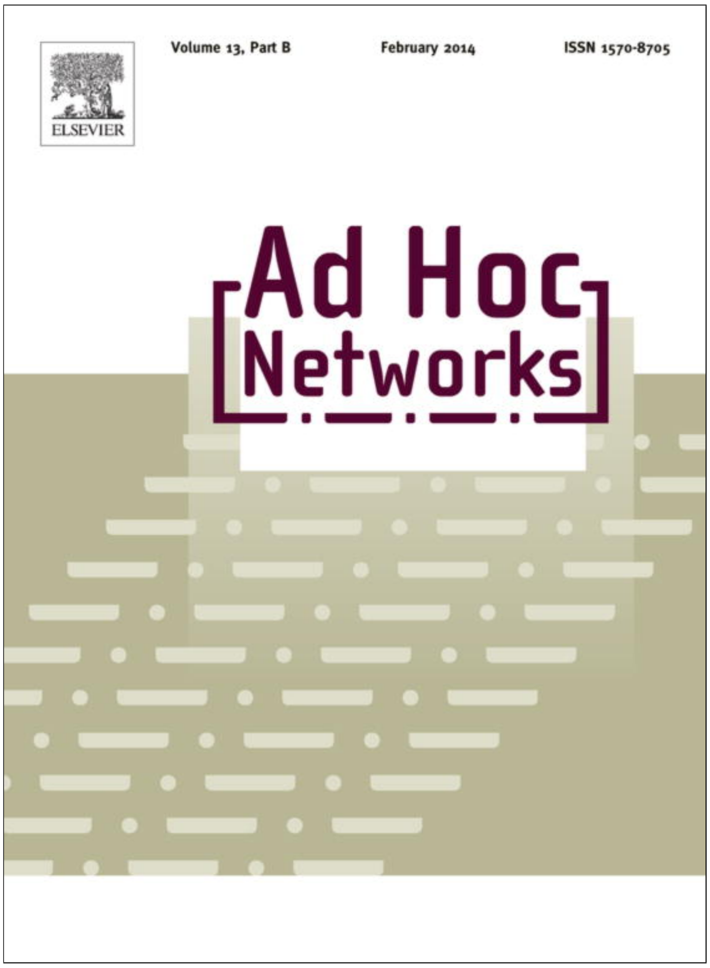

Volume 13, Part B
February 2014
ISSN 1570-8705
ELSEVIER
Ad Hoc Networks

Ad Hoc Networks 13 (2014) 321–335

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

RSSI-based relative localisation for mobile robots

Luis Oliveira [a,*], Hongbin Li [b], Luis Almeida [a], Traian E. Abrudan [a]

[a] IT/DEEC-FEUP, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
[b] State Key Lab. of Industrial Control Technology, Zhejiang University, China

ARTICLE INFO

Article history:
Received 6 September 2012
Received in revised form 7 March 2013
Accepted 10 July 2013
Available online 31 August 2013

Keywords:
Localisation
RSSI
Velocity
Positioning

ABSTRACT

In this work, we develop an anchor-less relative localisation algorithm aimed to be used in multi-robot teams. The localisation is performed based on the Received Signal Strength Indicator (RSSI) readings collected from the messages exchanged between nodes. We use the RSSI as a rough estimate of the inverse of distance between any pair of communicating nodes, and we claim that such estimates provide a coarse information of the nodes relative localisation that is still suitable to support several coordination tasks. In addition, we introduce a relative velocity estimation framework based on the RSSI measurements. This framework uses consecutive distance measurements and position estimates to provide the relative velocity vectors for all the nodes in the network.

To accomplish this, we propose using a Kalman filter and the Floyd–Warshall algorithm to generate smooth RSSI pairwise signal distance for all nodes. Then we use Multidimensional Scaling to obtain relative positions from the pairwise distances. Finally, due to anchor unavailability, relative positions are adjusted to reflect the continuous mobility by using geometric transformations, thus obtaining smoother trajectories for mobile nodes. This allows us to estimate velocity and to establish a correspondence between orientation in the physical world and in the relative coordinates system.

Additionally, we study the impact of several parameters in calculating the network topology, namely different approaches to provide a symmetric distances matrix, the period of the matrix dissemination, the use of synchronisation of the transmissions, and the filtering of the RSSI data. Experimental results, with a set of MicaZ motes, show that the period of matrix dissemination is the most relevant of the parameters, specifically with larger periods providing the best results, however, shorter periods are shown to be possible as long as the transmissions are synchronised.

1. Introduction

Multiple-robot systems can accomplish tasks that cannot be achieved individually. That is why a cooperating team of mobile robots, joining together to accomplish a common objective with no human intervention, is an interesting possibility. Sample applications include surveillance, exploration, manufacturing, and large volume transportation [1–3].

Consider a small team of robots working in a relatively small area. Each of the robots is equipped with identical basic platform and communication module so that they can communicate in a predefined channel and move in a coordinated pattern. For the sake of cost, robots may have different sensing or actuating components, which means that some tasks have to be accomplished by specific robots. When one special event is detected by one robot, it may have to notify another robot which is far from the event area, but equipped with specific actuating component, to deal with such event. For example, in a mine sweeping application, it is advisable to spread a team of robots with mine detecting capability and equip only a small portion of

* Corresponding author. Tel.: +351 220413325.
E-mail addresses: dee10023@fe.up.pt (L. Oliveira), hbli@iipc.zju.edu.cn (H. Li), lda@fe.up.pt (L. Almeida), tabrudan@fe.up.pt (T.E. Abrudan).

1570-8705/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.adhoc.2013.07.007

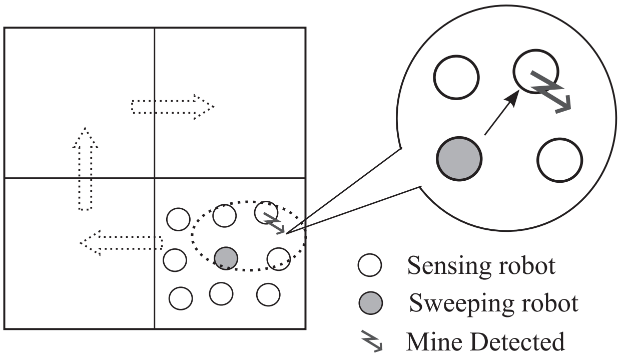


Fig. 1. A scenario of mine sweeping.

them with sweeping ability, thus reducing the cost of equipment. A typical scenario is depicted in Fig. 1. The mine field is divided into smaller areas according to the robots sensing range and the robots sweep the areas one by one with certain formation to guarantee coverage. When mines are detected, a robot with sweeping ability is informed to approach the specific spot. For both maintaining the formation and relocating the sweeping robot, relative positions have to be managed. In both cases, it is important to know the relative positions in order to make the decisions involving moving robots from one place to another. In some situations, a possible solution is to build an infrastructure that enables every robot to know its own absolute position. But, building infrastructure is costly and it is probably unavailable in urgent scenarios. For outdoors, GPS may be a possible solution; however, it only provides coarse-grained positions and it is satellite dependent, thus it is not available everywhere, such as in indoor spaces and street canyons. A possible solution, which is considered in our work, is to derive relative positions from local communication.

In our work, we are interested in the relative positions of the nodes, the accuracy in terms of absolute locations is not a concern. Therefore, we define proximity to a given node as a function of Received Signal Strength Indicator (RSSI), i.e. a node is closer than another one if the RSSI of its transmission is higher. Thus, instead of measuring the distance between nodes, we use RF signal strength as a measure of proximity, despite the coarse relationship between them, in the absence of a propagation model. A rough map of the nodes is produced with the coarse locations obtained using their signal proximity in terms of RSSI. Our method has the advantage of providing relative localisation based on the strength of the communication links, i.e. a pair of nodes that posses a good link are considered closer to each other, whereas nodes that have poor or no links are considered to be “far” apart from each other.

In order to work effectively with the RSSI, we implement a scheme for filtering and sharing the RF signal strength sensed in each node via the propagation of a signal space distance matrix. Then, we calculate the relative position of the nodes using MultiDimensional Scaling (MDS). Experiments show that signal strength information can be used to manage the relative positions of wirelessly connected nodes without central supervision.

This paper extends our previous work in [2] with:

1. A detailed and improved explanation of the relative localisation and the relative velocity estimation framework for mobile robots based on RSSI measurements.
2. A new experimental sensitivity analysis with respect to several configuration parameters.

This paper is organised as follows, Section 2 discusses the related work. Section 3 addresses the problem of sharing the signal space distance matrix. Section 4 addresses the problem of generating relative positions using the classical MDS algorithm and some data filtering techniques. Section 5 addresses the estimation of signal space velocity vectors. Section 6 describes a set of practical experiments to validate the robustness of the MDS algorithm under several conditions. Finally, Section 7 concludes the paper.

2. Related work

The location of the nodes in a network of mobile robots is an essential information required in order to put into practice a diversity of coordination algorithms, such as team formation and path planning. For example, in [4] the idea of using feedback laws to control multiple robots together in a formation is explored. In this work it was assumed that each robot has the ability to measure the relative position with respect to its closest neighbours. Also, in [5], the robots path is computed to ensure that the network partition never occurs during the robots motion, but the knowledge of global location (e.g. GPS) is assumed available at each robot. The work in [6] explores the sensor relocation in order to deal with sensor failure or respond to new events. Methods of finding redundant sensors and moving sensors to specific areas are proposed, assuming that sensors are placed into grids and global information is shared to support relocation planning. None of these works consider the practical position management of mobile robots or sensors.

Static sensor positioning has been widely investigated in recent years, where the dominant time-based techniques to obtain distance measurements include the Time of Arrival (ToA), the Time Difference of Arrival (TDoA), the Angle of Arrival (AoA), and the Time of Flight (ToF). Time of Arrival is a technique that is used to measure the time a given message needs to travel between the sender and several receiving nodes [7–9]. To do that, the sending node timestamps the message with global time and sends it. Then, the receiving nodes timestamp the message upon arrival and compute the message travelling time. Finally, knowing that the waves travel at the speed of light, the nodes can calculate the distance between themselves and the sender. However, due to the speed at which the waves travel, all the nodes clocks synchronised with very high precision, since a small timing error translates into a large distance error. Another popular method is TDoA. This method, similarly to ToA, measures the time a given message needs to travel between nodes, but unlike ToA it measures the time difference between receiving times. This method only requires the nodes performing the tracking to be synchronised, thus partially solving the global clock synchronisation issues [10–12]. Despite that, this method only brings advantages to situations where only some nodes can be easily synchronised to perform the tracking, such as the AoA. The AoA is a positioning technique that involves measuring the angle from which the received message comes from, thus obtaining the direction of the transmitter node [13,14]. This is done by means of antenna arrays, that can be more easily synchronised to use techniques such as TDoA to infer the direction of the incoming transmission. However, for mobile robotic applications, antenna arrays may be prohibitive in size. A technique that eliminates the need for global clock synchronisation is ToF [15]. This technique is also very similar to ToA in that it allows measuring the time a message needs to travel between communicating nodes. However, in order to remove the need for global clock synchronisation, instead of measuring the time of one-way trip, it measures the time that a message needs to go to the receiver and return to the transmitter. Despite that, since some local processing needs to be done on the receiver before sending the reply, the processing time has to be very well known, thus it should be done in hardware. Adding to that, since the ranging operation is between two units, it needs a long time to range several units, thus it may not accommodate fast moving robots.

Another method to perform ranging between communicating nodes is to measure the Received Signal Strength Indicator (RSSI). As the name implies, RSSI-based methods, obtain range estimation from the strength of the received RF signal [16]. In open space and without interference there is a relationship between RSSI and distance. However, in the presence of interference, reflection, and refraction, this relationship can be easily destroyed. Despite that, methods other than RSSI-based ranging require either specialised and expensive hardware, or require global time synchronisation, which may be difficult to attain. Therefore, if the application requires a coarse localisation, only, either for navigation or topology estimation purposes, the RSSI can still be useful. Several RSSI-based methods rely on a priori channel measurements [17,18]. However, those may be unavailable or unreliable due to lack of previous knowledge on the environment or severe changes that might have affected it. Other methods perform online channel estimation, frequently based on measurements between anchor nodes [19,20]. However, these are incompatible with unknown environments, or require sensors beyond communication devices [18].

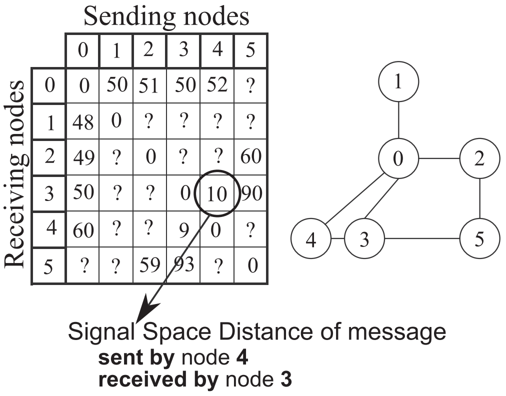
Once ranging data is collected, several techniques can be used to generate sensor positions [21–25]. One of such techniques [26,22] uses an improved versions of MDS, i.e. MDS-MAP(P) and MDS-MAP(C), to obtain node positions based on the distances between nodes. MDS-MAP(C) is a method that after running classic MDS uses an extra step to transform the relative positions to a global frame using anchor node information. MDS-MAP(P) applies MDS-MAP(C) to a set of nodes up to a maximum of n hops. Then, each of the generated maps is stitched together with the neighbours until a map of the full network is produced. Finally relative positions are adjusted to a global frame using anchor node information. Also in the same work, the authors present an extended version of both algorithms, MDS-MAP(P,R) and MDS-MAP(C,R). In these variants a final refinement step, using least-squares minimisation to make the distances between neighbouring nodes better match the provided measured distances. The work in [27] uses SISR (Snap-Inducing Shaped Residuals) to perform an iterative refinement to the positions returned by MDS-MAP(C,R). This method is able to tolerate ranging errors by de-emphasising bad data, and was shown to localise nodes in non-convex topologies. These variations of MDS, despite performing better than the regular MDS, are not worth the extra computational cost for our application, since we target small size networks with just a few hops. In order to improve results under unknown line-of-sight/non-line-of-sight (LOS/NLOS) conditions and scarce ranging information [28], uses another variant of MDS based on Weighted Least Squares algorithm, whose weights are assigned according to the reliability of the ranging measurements. The work in [29] provides a complete theoretical analysis of node localisation using MDS. However, most of the existing work either assumes known position of anchor nodes, which is unavailable or unnecessary in a scenario like ours, or only provides simulation results.

Adding mobility to sensor nodes appears to make localisation more complex and uncertain. The work in [30] employs Monte Carlo Localisation methods to improve accuracy. Nodes that know their own location – called seeds – and nodes with unknown location form a network, where at least one kind of sensors is moving. Despite the improved localisation, Monte Carlo solutions are computationally expensive, consequently very hard to implement in real-time. Another approach to mobile robots localisation can be found in [31], however each robot is equipped with a range sensor to identify distance to other robots and odometry, thus increasing the cost.

Our paper differs from the previously referred work in several aspects. Firstly, we investigate the problem in which no anchor node exists. We use MDS [32] and RF signal strength readings (only) to compute approximate relative positions within small teams of mobile nodes (approximately up to 10), with the purpose of managing the nodes mobility. Thus, we are not interested in accurate physical localisation. We claim that such approximate relative positioning system is sufficient for a set of navigation purposes in particular to drive certain nodes to the vicinity of other ones. In addition, we provide an experimental sensitivity analysis with respect to the impact of several configuration parameters in the MDS algorithm. Finally, we present a relative velocity estimation framework, also based on the RSSI measurements alone.

3. Building the signal space distance matrix

Due to high mobility together with nodes joining and leaving the team at run time, we need a scheme to keep track of the connectivity information within the group. The authors of [33] proposed a concept called the connectivity matrix, which enables every mobile node to keep a global vision of the team composition and network topology.



Receiving nodes \ Sending nodes	0	1	2	3	4	5
0	0	50	51	50	52	?
1	48	0	?	?	?	?
2	49	?	0	?	?	60
3	50	?	?	0	10	90
4	60	?	?	9	0	?
5	?	?	59	93	?	0

Fig. 2. Signal space distance matrix: The matrix represents the distance between every pair of nodes in the network.

However, as [2] points out:

1. The binary representation of a link state provides insufficient information on the link quality, and is prone to instability.
2. This method requires synchronisation in a strict TDMA fashion.
3. Multi-hop communications require one extra TDMA slot per hop, thus leads to substantial time slot reassignments due to topology changes.

Therefore, in order to have a more accurate link state representation we propose a signal space distance matrix, stemming from the Extended Connectivity matrix in [2], and drop the TDMA-based synchronisation.

3.1. Signal space distance matrix

The extended connectivity matrix is defined in [2]. Instead of binary information, it contains the RSSI values of the received packets (in dBm) for every link. However, since in this work the raw RSSI values are not used directly, we redefine this matrix using signal space distance (**D**), that represents the “distance” to a maximum value of RSSI ($\mathtt{RSSI}_{\mathtt{max}}$) as in (1). $\mathtt{RSSI}_{\mathtt{max}}$ is an offset that can be used to define distance according to the offset of the transceiver being used, i.e. if the data obtained by the transceiver has no offset then $\mathtt{RSSI}_{\mathtt{max}} = 0$.

$$\mathbf{D}^k(i,j) = \begin{cases} \mathtt{RSSI}_{\ \mathtt{max}} - \mathtt{RSSI}_{\ i,j}, & \mathtt{LQI} \geqslant \mathtt{LQI}_{\ \mathtt{threshold}} \\ ?, & \text{packet lost or } \mathtt{LQI} < \mathtt{LQI}_{\ \mathtt{threshold}} \end{cases} \qquad (1)$$

Each mobile node maintains an signal space distance matrix **D**. $\mathbf{D}^k$ is the matrix stored in node k. For each row i, $\mathbf{D}^k(i)$ stores the signal space distance readings measured by node i, or “?” representing the absence of such measurement, including for $i = k$. Fig. 2 shows an example of an signal space distance matrix with 6 nodes and their wireless links represented on the right side.

3.2. Absence of synchronisation

In our system model, we do not rely on clock synchronisation. It is generally recognised that in lightly loaded communication conditions, the cost of retransmissions is lower than the cost of preventing them. For example, the MicaZ motes communicate in 2.4 GHz with a transmission rate of 250 Kbps, thus, a packet of 300 bytes needs less than 2 ms to be transmitted. Since the usage of the bandwidth is only 4%, with a long enough message broadcast period, e.g. 500 ms, the probability of message collision for 10 nodes is negligible.

3.3. Updating algorithm

The signal space distance matrix is filled in by all nodes in a distributed way. Each node contributes with the signal space distance values of the messages it receives, and fills its own line. Then, the matrix is broadcast to the other nodes, that in turn merge the received matrix with their own and disseminate it closing the cycle. The merging of the matrices must be done with care, though. In order to avoid keeping stale data and replacing fresh data with old data, we use the following control variables:

Algorithm 1. Updating algorithm for node k signal space distance matrix, $\mathbf{D}^k(i,j)$, with time-stamps and sequence control.

```
Receive phase:
IF node $k$ receives the expected $\mathbf{D}^w$ then
  for all $i \neq k$ do
    IF $\mathtt{Sequence}^w(i) > \mathtt{Sequence}^k(i)$ then
      $\mathbf{D}^k(i) = \mathbf{D}^w(i)$
      $\mathtt{Sequence}^k(i) = \mathtt{Sequence}^w(i)$
    end if
  end for
  if ($\mathtt{LQI}_{w \to k} > \mathtt{LQI}_{\mathtt{Threshold}}$) then
    $\mathtt{SamplesBuffer}(w)$.add ($\mathbf{D}_{w \to k}$, $\mathtt{LQI}_{w \to k}$,
 localtime)
  else
    $\mathtt{SamplesBuffer}(w)$.add (?, $\mathtt{LQI}_{w \to k}$,
 localtime)
  end if
end if
Broadcast phase:
if timer triggered for node $k$ to broadcast $\mathbf{D}^k$ then
  for all $j$ do
    $\mathbf{D}^k(k,j)$ = Filter(($\mathtt{SamplesBuffer}(j)$))
  end for
  $\mathtt{Sequence}^k(k) = \mathtt{Sequence}^k(k) + 1$
  Broadcast ($\mathbf{D}^k$, $\mathtt{Sequence}^k$)
end if
Eliminate outdated data phase:
if timer triggered for eliminating outdated data
 then
  for all $i \neq k$
    if $\mathtt{localtime} - \mathtt{Time}^k(i)$
 $> \mathtt{OutdateThreshold}$
      $\mathbf{D}^{k(i)} = \{?\}$
    end if
    SamplesBuffer (i). removeOutdated ()
  end for
end if
```

1. Local time-stamps, indicating the freshness of the data.
2. Sequence numbers, indicating which is the fresher one, between the same line of information in two matrices.

The time-stamps are organised in a vector with each element controlling the age of one signal space distance matrix line. A time-stamp is updated together with the respective line. Then, every receiver keeps several signal space distance samples per sender in a circular buffer for filtering purposes. These samples also have time-stamps associated with each one of them, which are also updated whenever the samples are overwritten. The sequence numbers are organised in a vector, each element corresponding to a line in the signal space distance matrix. Each node increases its sequence number right before sending it together with the updated signal space distance matrix. Note that the vector of sequence numbers concerns all lines in the matrix, each value given by the respective producer node. The referred data structures are shown in Fig. 3. All nodes expect packets from each other in each round and the sequence number allows them to reject older lines. When packets are received with an LQI which is above a certain threshold, the corresponding signal space distance value is used and inserted in the respective samples buffer. On the other hand, if the LQI is below that threshold, the signal space distance value is discarded. Thus, the samples buffer stores the latest signal space distance and LQI accepted readings together with a local time-stamp, tolerating a configurable time without receiving messages, before declaring a link disrupted. When it is time to broadcast its own signal space distance matrix, each node filters the signal space distance readings in the samples buffer and stores the filtered value in the corresponding row, updates its sequence number and transmits the matrix. In addition, each node goes periodically throughout the matrix and sample buffer, removing the values whose age is greater that a given threshold. The threshold value should be small enough to cope with the team dynamics and large enough that tolerates a few missing packets. Algorithm 1 summarises our distributed algorithm to update the signal space distance matrix.

3.4. Filtering the proximity measurements

There are two main non-idealities in our system. The first one, is occasional packet loss, due to the unreliability of wireless communication. This also poses a problem for mobile nodes management given the difficulty in distinguishing node absence from packet loss. The second non-ideality is the instability of the RSSI readings that propagates to signal space distance. Two nodes, even placed in fixed positions without human activity or electromagnetic interference, receive fluctuating RSSI readings from each other due to complex dependence with several parameters of the medium. For a group of mobile nodes, this instability becomes even harder to handle. Some previous studies use techniques like averaging, frequency and space diversity, and signal modelling to counteract the RSSI instability. For the occasional packet losses, as explained before, each node uses a sliding window for packet buffering (see *SamplesBuffer* in Algorithm 1). In each broadcast period, if a packet is received, the signal space distance value is put into the respective sliding window. If, on the other hand, the packet is lost, nothing is put in the window being the

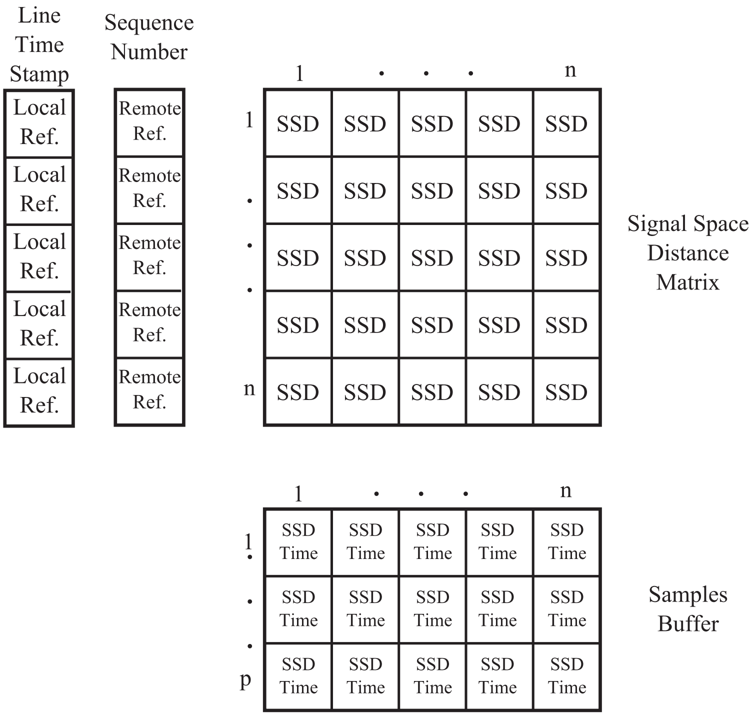


Fig. 3. Control variables to update the matrix in node k: Top (from left to right) vector containing the local age of each line of the matrix, vector containing the sequence number of each line of the matrix, the signal space matrix; Bottom buffer containing the samples of signal space distance.

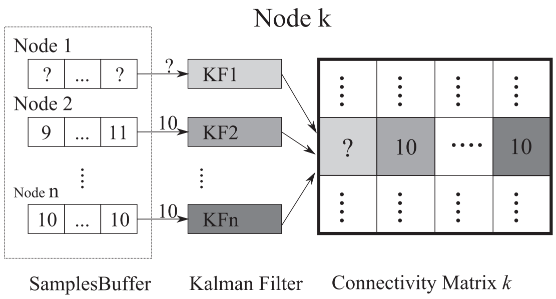


Fig. 4. The filtering process for **D**: The samples in the buffer are filtered and used as input to the Kalman filter; the output is written in line k of $\mathbf{D}^k$.

old values discarded later. Then, before broadcasting the signal space distance matrix, the filter is applied. For this work we propose to compare two different filters:

1. Using a 3 sample sliding window where all non-zero samples are taken into account.
2. Using a 5 samples window with the following rejection rule:
   (a) If only one non-zero value exists, it is used as is.
   (b) If two exist, the larger is rejected.
   (c) If more than two exist, the highest and the lowest are rejected and the others are averaged.

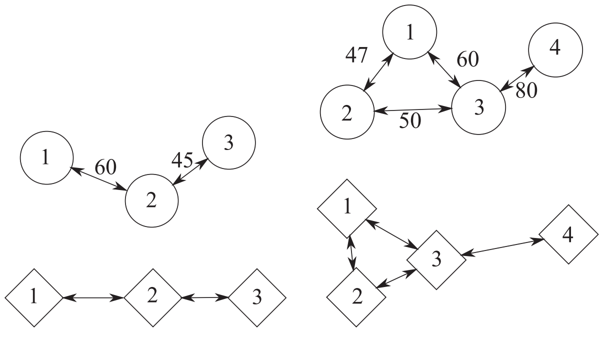
In order to track the signal strength proximity over time, a scalar Kalman filter [34] is employed for each element $d_{i,j}$ of the matrix **D**, Eq. (1). The time evolution of $d_{i,j}$ is modelled as a first-order autoregressive process. The corresponding prediction rule is given by Eq. (2), where k is the discrete-time index, w is the zero-mean white Gaussian noise of the process with standard deviation σ_w, and describes the shadow fading process.

$$d_{i,j}(k) = (1-\epsilon)d_{i,j}(k-1) + w(k), \tag{2}$$

The small value $\epsilon \in (0,1)$ introduces correlation between successive true states and ensures the wide sense stationary property of the process. ϵ depends on the channel coherence time, as well as on the broadcast period of the signal strength proximity matrix. The measurement equation is presented in Eq. (3), where $\upsilon(k)$ is the white Gaussian measurement noise with standard deviation σ_υ, uncorrelated to $w(k)$.

$$\delta_{i,j}(k) = d_{i,j}(k) + \upsilon(k), \tag{3}$$

The result is then used as the signal space distance value for that node. The filtering process is illustrated in Fig. 4. As shown in [2] this solution significantly improves the results by smoothing the measured data, and reduces undesired fluctuations.



(a) A case of 3 nodes (b) A case of 4 nodes

Fig. 6. Relative positions in physical space and in signal strength space: Top – real network; Bottom – signal space network.

4. Generating signal space relative positions

Using the communication scheme of the previous section, a small group of mobile nodes can share both topology and signal space distance information for each pair of nodes, in the form of the signal space distance matrix. Based on those, an approximate global vision of the whole relative positions can be generated in each node by the process depicted in Fig. 5, which will be hereinafter explained.

By deeming nodes with strong RF connection to be neighbouring nodes and nodes with weak connection to be further apart, the relationship between relative positions in physical space and proximity in signal strength space can be illustrated as in Fig. 6. There, circles denote the physical nodes positions and bricks denote positions in signal strength space. As we may notice in Fig. 6, the proximity in signal strength space only depends on the signal strength and not directly on the physical distance. Nevertheless, this information is still sufficient for navigation purposes, for example, to bring a given robot closer to another one, and to allow a robot to know his neighbours.

4.1. Approximating missing measurements

In practice, collecting all pairwise distances is often impossible. This is the case of wireless mobile nodes, since some links can be broken due to mobility or limited communication range. When this happens, the signal space distance matrix will contain empty values – represented with "?" – and the classical MDS algorithm cannot be directly applied. Several techniques have been proposed in the literature in order to solve this problem, e.g. Map Stitching [35,36], Iterative MDS [37], non-linear regression (NLR) [38], and the extension to MDS proposed in [39]. Despite that, Map Stitching and Iterative MDS are not able to recover the correct topology of weakly connected networks such as in Fig. 7, a situation that can occur in formation

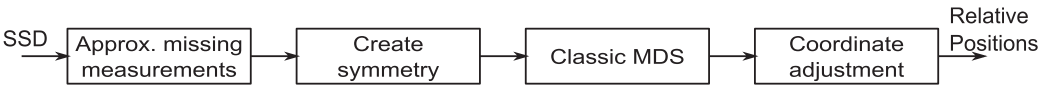


Fig. 5. Generating positions from the signal space distance.

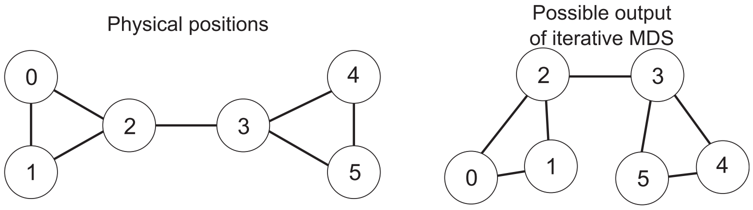


Fig. 7. Joining weakly-connected networks using iterative methods: There is nothing pushing disconnected nodes apart resulting in a very distorted topology.

control of small mobile teams. Moreover, the Iterative MDS and NLR both need to perform iterations, consequently, due to the dynamics of mobile robot, they are undesirable. Finally, the extension to MDS assumes that there are two groups of nodes: (1) nodes forming a fully connected network; (2) nodes which are fully disconnected amongst each other, but that are able to communicates with each node in the first group. This assumption, however, is not valid for mobile networks, whose dynamics are high. For these reasons, we use Classical MDS [32] plus the Floyd–Warshall [22] algorithm to estimate missing distances, because even with only 50% of links, the resulting positions are acceptable [2]. In addition, the obtained topology is more accurate than with the Map Stitching and Iterative MDS approaches (Fig. 7).

We herein assume a connected network (despite possibly not fully linked), meaning that there exists at least one route between any pair of nodes, consequently the network is not partitioned. Let E denote a route between i and j, which contains several links, and let the pair of nodes a and b be the extremes of a generic link in E. We thus define $\mathbf{F}^k(i,j)$ according to expression (4), approximating the distance between two nodes that are not directly connected, with minimum accumulated signal space distance.

$$\mathbf{F}^k(i,j) = \begin{cases} \mathbf{D}^k(i,j) & \text{if } i \text{ and } j \text{ are linked} \\ \min\left(\sum_{\forall(a,b)\epsilon E} \mathbf{D}^k(a,b)\right) & \text{if } i \text{ and } j \text{ are not linked} \end{cases} \tag{4}$$

As shown in Fig. 8, the physical distance between two indirectly connected nodes is probably smaller than the minimum accumulated distance of a connection route. For example: $\texttt{Dist}_{13} < \texttt{Dist}_{12} + \texttt{Dist}_{23}$. This introduces another source of deformation in the nodes relative positioning with respect to their physical position. However, as we said before, we are just targeting for relative position estimates for navigation purposes and the referred deformation should not be an obstacle to that purpose. On the other hand, this distance estimation is deterministic and easy to compute, enabling a smooth position estimation in scenarios of moving nodes. Similarly to what is explained in [2] we use the Floyd–Warshall algorithm for computing the shortest signal distance for every pair of nodes. After completion, this algorithm provides a complete matrix of pairwise signal space distances. Our distance approximation approach requires an additional time complexity of $\mathcal{O}(n^3)$ due to the Floyd–Warshall algorithm. We consider this affordable for small teams of mobile nodes in which n is typically ten or less.

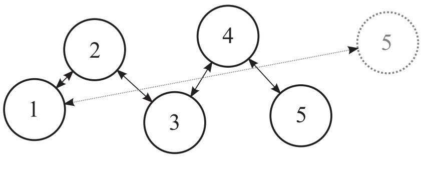


Fig. 8. Approximating the distance for missing SSDist measurement.

4.2. Creating a symmetric matrix

Since pairwise distances should be reciprocal, one of the Multidimensional Scaling technique requirements is the symmetry of the signal space distance matrix. However, due to communication uncertainty, slightly different transmission power in different nodes, non-omnidirectional antennae, etc., the signal space distance matrix is seldom symmetric. That being said, in order to create and feed a symmetric distance matrix to the MDS algorithm, we propose the following options:

- Use one of the triangles of the matrix, either upper or lower, and replace the other one.
- Use the mean between the upper and lower triangles of the matrix (i.e.: mean between the signal space distance values of both directions in each link).
- Use the maximum signal space distance value between the upper and lower triangles of the matrix (i.e.: minimum RSSI value between the two directions in each link).
- Use the minimum signal space distance value between the upper and lower triangles of the matrix (i.e.: maximum RSSI value between the two directions in each link)

To the best of our knowledge, a study on the impact of such options on MDS performance has not been carried out before, thus we assess the problem in the experiments that are shown later on.

4.3. Multidimensional Scaling

As proposed in [2], we use MDS [32] to compute the relative positions. This choice was made due to the relative simplicity to implement MDS, as well as to evaluate the ability of MDS to create a topology assessment. MDS is a technique used in multivariate analysis that transfers a known $n \times n$ matrix ($\mathbf{A}$) of dissimilarities to n points of an m-dimensional Euclidean space in such a way that the pairwise distances between points are compatible with

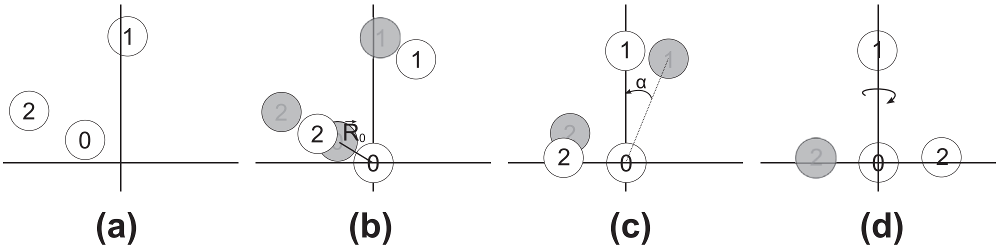


Fig. 9. Adjusting coordinates: (a) Positions given by MDS; (b) Positions after shifting node 0 to origin; (c) Positions after rotation of node 1; and (d) Positions after flipping node 2 to the right plane.

the dissimilarities matrix. Being $[\delta_{ij}]_{n\times n}$ the matrix with pairwise distances, this algorithm can be used to find an $\mathbf{X} = [x_{ij}]_{n\times m}$ matrix of approximate positions.

Our goal is to avoid using a channel model (e.g. log distance path loss model) that would require calibration based on measurements, i.e. additional parameters to be estimated. The mapping from the physical distances to signal space distances is a log function, and this defines our new disparities used in the MDS [32, Ch. 9]. This is a reasonable assumption, as we are only coarse interested in relative positions, and not in accurate physical positions of the nodes.

4.4. Adjusting the relative coordinates

So far, we discussed the relative position of a team of mobile nodes with no physical anchor. However, for the MDS algorithm, a small perturbation in the distances matrix would bring totally different results for the coordinates $\mathbf{X}$. One of the causes for such behaviour is the way MDS sorts out certain ambiguities that are inherent to the relative localisation process, e.g. eigenvector switching in the subspace selection process causes map flips. Since the nodes position is only recovered up to rigid motion, orientation of the team cannot be determined just with pairwise distances, neither can the symmetry relationships. To obtain relative position estimates that vary smoothly, we carry out the following adjustments of the coordinates provided by the MDS (considering only the result presented in 2D space, i.e. $m = 2$). Let $\mathbf{R} = [r_{ij}]_{n\times 2} = (\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_{n-1})$ denote the coordinates determined with MDS (Fig. 9a) and $\mathbf{S} = [s_{ij}]_{n\times 2} = (\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{n-1})$ denote the final coordinates (Fig. 9d). We consider the three nodes with the smallest IDs as being local references (herein referred to as 0, 1, and 2).

The coordinates adjustment includes shift (Fig. 9b), rotation (Fig. 9c) and reflection (Fig. 9d) so that node 0 is at the origin point (0,0), node 1 on the positive y-axis and node 2 on the right half-plane. Thus, we first let $\forall_{0\leqslant i<n}$ $\mathbf{t}_i = \mathbf{r}_i - \mathbf{r}_0$ as in Eq. (5), obtaining $\mathbf{T}$ where node 0 is in the origin.

$$\begin{aligned}\mathbf{T} &= (\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{n-1}) \\ &= (\mathbf{r}_0 - \mathbf{r}_0, \mathbf{r}_1 - \mathbf{r}_0, \mathbf{r}_2 - \mathbf{r}_0, \ldots, \mathbf{r}_{n-1} - \mathbf{r}_0)\end{aligned} \tag{5}$$

Then, we compute the clockwise angle α from vector $\mathbf{t}_1$ to y-axis, and rotate all nodes α around the origin as in Eq. (6) deducing the intermediate positions $\mathbf{Y}$.

$$\mathbf{Y} = (\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{n-1}) = \mathbf{T} \times \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \tag{6}$$

Finally, we check if $\mathbf{y}_2$ is on the right half-plane, i.e. if node 2 has a positive x-coordinate. If so, $\mathbf{S} = \mathbf{Y}$, else we reflect $\mathbf{Y}$ over the vertical axis as in Eq. (7).

$$\mathbf{S} = \begin{cases} \mathbf{Y}, & \text{if } \mathbf{y}_2 \text{ is in the right plane} \\ \mathbf{Y} \times \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, & \text{otherwise} \end{cases} \tag{7}$$

5. Estimation of signal space velocity based on RSSI

In this section, we show the importance of the coordinates adjustment and filtering for scenarios with mobility. In addition, we provide a method to estimate the relative velocity of the nodes in the signal strength space based on the difference of consecutive relative positions. Note that since we estimate signal space velocity from the positions, using the former to further improve the latter would not bring benefits.

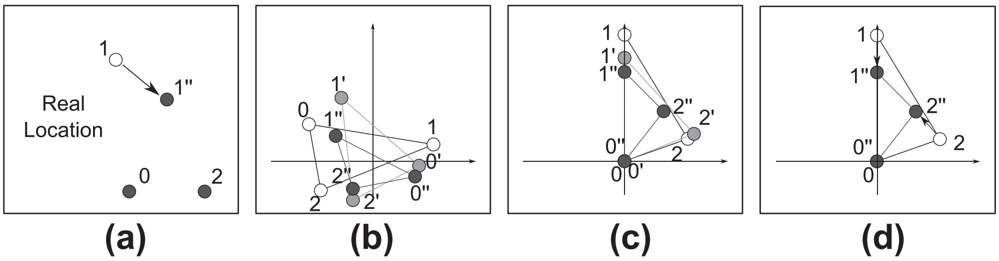


Fig. 10. Smoothing the moving trajectory: (a) Physical positions; (b) MDS; (c) MDS with coordinate adjustment; and (d) MDS with coordinate adjustment and Kalman filter.

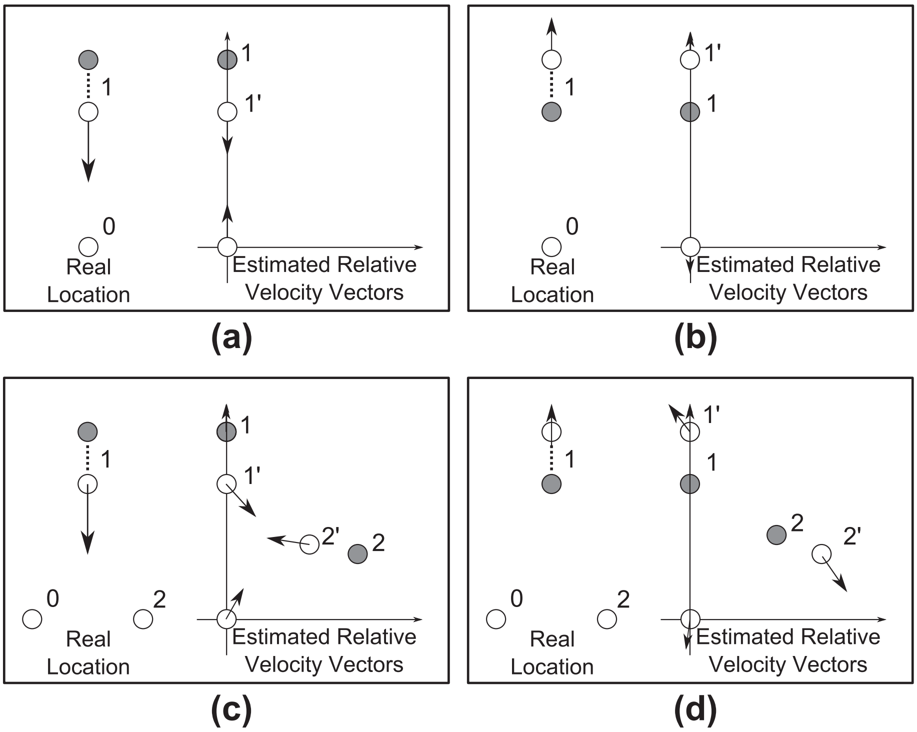

1
1'
0
Real Location
Estimated Relative Velocity Vectors
(a)
1
1'
0
Real Location
Estimated Relative Velocity Vectors
(b)
1
1'
2'
2
0
2
Real Location
Estimated Relative Velocity Vectors
(c)
1
1'
2
2'
0
2
Real Location
Estimated Relative Velocity Vectors
(d)

Fig. 11. Observing relative mobility: (a) Node 1 approaches node 0; (b) Node 1 moves away from node 0; (c) Node 1 approaches both node 0 and 2; and (d) Node 1 moves away from both node 0 and 2.

5.1. Impact of data filtering and coordinate adjustment on trajectory smoothing

Consider a simple case with 3 nodes in which one of them approaches the other two as depicted in Fig. 10. Fig. 10a depicts the physical position of nodes 0, 1 and 2. Node 1 moves slowly to the dashed position 1”. Three cases where considered, pure MDS, MDS with coordinate adjustment, and MDS with coordinate adjustment and Kalman filter. Fig. 10b shows the dynamic relative position results of pure MDS, before applying the coordinates adjustment, and without the Kalman filter. The white circles $(0,1,2)$ denote the initial position of the nodes, and the grey circles $(0',1',2')$ denote a set of typical intermediate positions. Final positions are illustrated by the black circles $(0'',1'',2'')$. Due to the absence of a fixed reference, the calculated positions are unstable. Even when the nodes are stopped, small fluctuations in the RSSI readings cause MDS to generate totally different results. Fig. 10c shows the result after applying coordinates adjustment, but still without the Kalman filter. The units trajectory is now more consistent but occasional sudden jumps are still observable. Finally, Fig. 10d shows the results for the case when a Kalman filter is used to track the RSSI readings. We obtain smoother moving trajectories and no position jumps or flips are observed. The signal distances provided to MDS change smoothly, thus producing smooth variations of the relative positions.

5.2. Estimating signal space velocity

Smoother trajectories allow us to assess the possibility of estimating the relative velocity of the nodes in the signal strength space. Our proposal is based on the difference of consecutive relative positions and distances which can be very useful to several applications, such as formation control. For each new estimate of relative positions, we can compute the difference of the consecutive pairwise distance measurements for each pair of nodes $(d_{ij} - d_{ij}^-)$ where, d_{ij} = F (i,j). Then, compute a unit vector $\mathbf{s}_i - \mathbf{s}_j/|\mathbf{s}_i - -\mathbf{s}_j|$ pointing from i to j. The signal space velocity of node i is estimated summing the relative movements to all other nodes and dividing by the transmission period T_t as in Eq. (8).

$$\mathbf{v}_i = \frac{1}{T_t}\sum_{j=1,\ldots,n,j\neq i}(d_{ij} - d_{ij}^-)\cdot\frac{\mathbf{s}_i - \mathbf{s}_j}{|\mathbf{s}_i - \mathbf{s}_j|} \qquad (8)$$

Fig. 11a and b shows the case of two mobile nodes. White circles represent the current position, grey circles represent the previous position. On the left hand side of the plot, the physical positions are depicted, with the arrows representing the signal space velocity vectors. On the right hand side of the plot, the results in signal strength space are shown, with the arrows representing the relative velocities. Note that although node 1 is not moving physically, the obtained signal space velocity vectors indicate that both nodes are approaching each other or moving apart, which is correct given the relative nature of the determined positions and velocities. Fig. 11c and d shows another example with three mobile nodes. When node 1 moves closer to, or away from nodes 0 and 2, the signal space velocity vectors of all three nodes correctly indicate a convergence or divergence among them. The signal space velocity estimation is an important result of this paper, because it allows obtaining information on orientation from

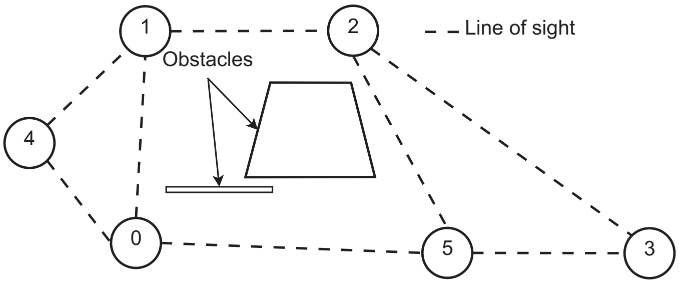


Fig. 12. Node distribution.

the relative positions estimates. When node a wants to move closer to node b or group C of several nodes, it can perform a tentative physical move and obtain feedback from the corresponding signal space velocity $\mathbf{v}_a$, and thus adjust its orientation accordingly to move towards the derived target location.

6. Experimental results

In this section, we describe an implementation of the proposed relative localisation algorithm using Crossbow's MicaZ motes. MicaZ motes communicate in 2.4 GHz IEEE 802.15.4 running TinyOS 1.1.15 operating system. We used the RSSI and LQI values measured at each packet reception.[1] The program running on all nodes was identical except for the unique node ID. The LQI values can range between 0 and 255, and according to our observation, they were typically above 100 when two nodes showed a good link state, and dropped dramatically below 70 when nodes showed a poor link with frequent errors and packet losses. For the purpose of RSSI measurements we considered only packets received with $LQI \geqslant 100$. For sensing data retrieval, an arbitrary node was connected to an MIB600 board that forwarded the data to a host computer via an Ethernet TCP/IP interface.

6.1. Sensitivity to parameter selection

The main objective of this section is to validate the proposed RSSI-based relative localisation for supporting collaborative behaviours in small teams of mobile robots.

We assess the impact of the choice of communications period, of the use of synchronisation among the nodes, and of data sampling and selection. For this purpose, we carried out several experiments in which we set the transmit power of six crossbow MicaZ motes to −10 dBm. Their typical RF range with the original antenna in an indoor lab is approximately 8–10 m. We conducted our test in a 5 m × 5 m area in which the RSSI reading ranged approximately between −48 dBm and −10 dBm and considered that $\mathtt{RSSI}_{\mathtt{max}} = 0$. The nodes were placed according to the diagram presented in Fig. 12, sending a periodic transmission with the sensing data as referred in the previous sections. A Matlab program running Java methods was designed to get the sensing data from the programming board via the TCP/IP port, capturing the information to a file for offline processing. Note that since node 0 was connected to the programming board. The results represent its perspective of the network. Every node can compute its own relative positions coordinate system. We ran four experiments, capturing 300 samples in each of them, two using a transmission period of 500 ms and the other two with a period of 100 ms, both with and without synchronisation. For the synchronisation algorithm we used the Adaptive Time Division Multiple Access (A-TDMA) algorithm [42,43]. The offline processing uses the data collected in the several experiments. A sliding window filter was applied to the data. First, a 3 sample window was used, where all non-zero samples were taken into account. Then, a 5 samples window was used with the previously discussed rejection rule. The Kalman filter was applied to the RSSI data, and then, the positioning with classical MDS algorithm was used, complemented with the Floyd–Warshall algorithm. The methods in Section 4.2 were used to obtain a symmetrical matrix. Finally, the nodes coordinates were adjusted, as described in Section 4.4. Note, however, the processing was done offline in order to carry out the analysis. As we will show hereinafter, the relative positioning algorithm can run online.

Fig. 13 shows the results of the position estimates from the perspective of node 0, where the dots represent single estimations, and the ellipses characterise the whole sample for each individual node. In particular, the ellipses show the standard deviation of the localisation errors along the main axes. In order to simplify the visualisation of the results, the plots show the case in which a symmetric pairwise distances matrix was achieved using the minimum signal space distance of the two directions in each link. Nevertheless, the results achieved with the other options in Section 4.2 were very similar.

6.1.1. Results on the transmission period

We considered two different periods for the nodes to broadcast their matrix, namely 100 ms and 500 ms. The results are showed in Fig. 13a and b for 500 ms and Fig. 13c and d for 100 ms. The former case presents higher precision, with significantly smaller standard deviation. On the other hand, it is less reactive than the latter case, in which changes in the physical topology are reflected in the matrix and distributed among the nodes five times

[1] For a comprehensive understanding of RSSI and LQI values provided by CC2420 [40], see [41].

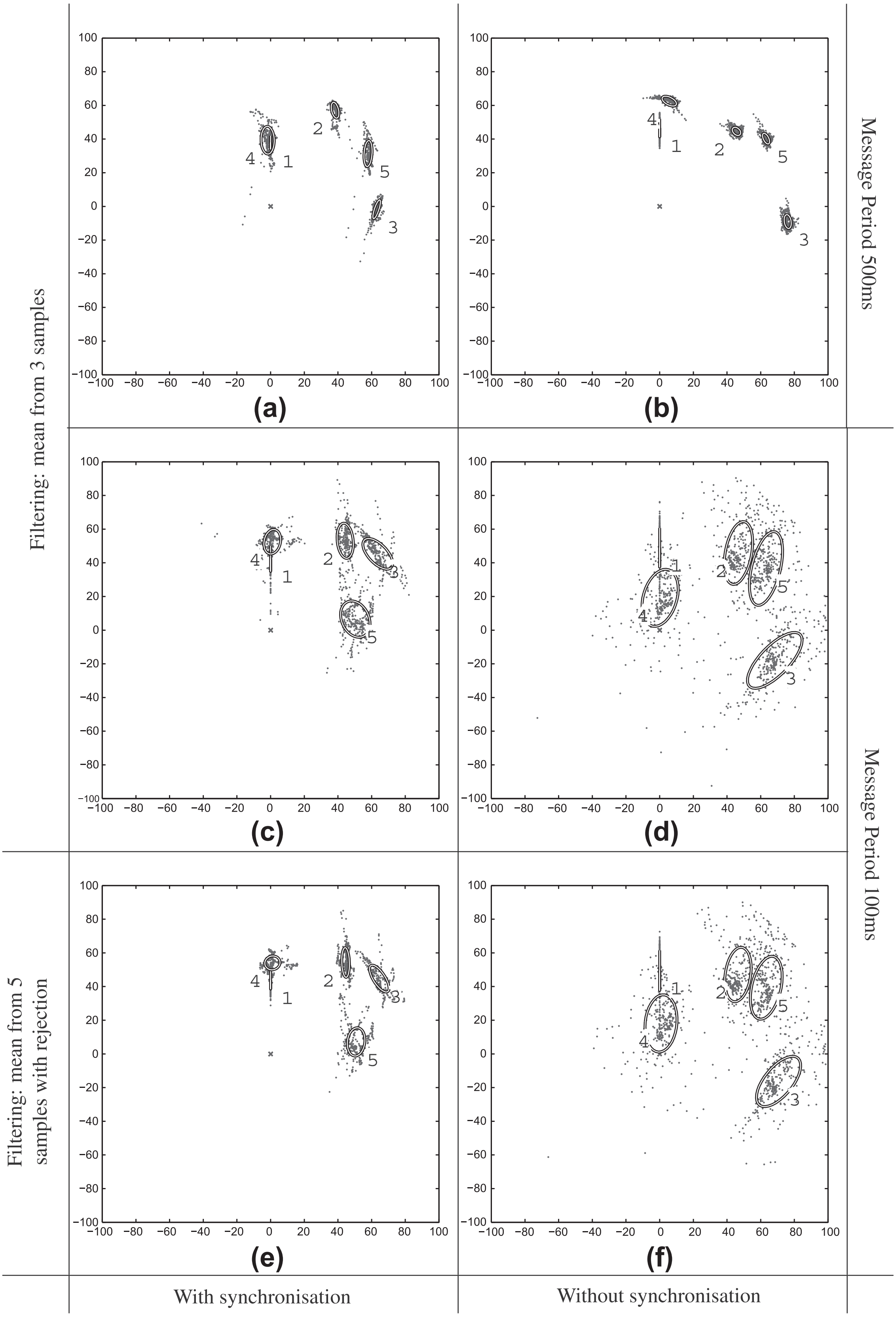


Fig. 13. Experiments concerning MDS: Position estimates using different sliding window filters, transmission periods, with and without synchronisation.

faster. The difference in standard deviation may be explained by a lower residual probability of collisions and interferences with longer periods, in the Adaptive-TDMA synchronisation.

6.1.2. Results on the use of synchronised transmissions

Despite using short messages (a 6 × 6 byte matrix, a 6 × 1 byte ageing vector, a message header, and message tail in a total of 55 bytes, thus with a low medium occupancy), the experiments show that the absence of synchronisation can cause a strong degradation in the quality of the localisation. The degradation is caused by the higher probability of collision with the other nodes and interference with other transmitters in the same band, e.g., a WiFi network was operating in the same area. This effect may be noticed by comparing Fig. 13c and e with Fig. 13d and f, respectively, where a transmission period of 100 ms was used. On the other hand, when the transmission period increases, the probability of collision reduces and the impact of synchronisation becomes less significant. This is shown in Fig. 13 a and b using a transmission period of 500 ms.

6.1.3. Results on the use of different sample window sizes and sample selection strategies

Comparing the results in Fig. 13c and d with those in Fig. 13e and f, obtained by using the same transmission period, we may notice the performance in terms of standard deviation slightly better when using 5 samples, compared to the case where 3 samples were used.

6.1.4. Testing the use of different approaches to produce a symmetrical matrix

In our experiments none of the different approaches tested showed a significant improvement over another one. Thus, when the transmission power of all nodes is approximately equal, as well as their antennas, the differences of the two halves of the matrix are not significant. Consequently, in order to use MDS, the signal space distance matrix can be considered symmetrical and, as such, the amount of transmitted data can be reduced from n^2 to $n \times (n-1)/2$, by transmitting only half of the matrix. This improvement may not seem significant since the complexity remains $\mathcal{O}(n^2)$, however, it halves the amount of data that needs to be transmitted.

6.1.5. Final considerations

Based on our experiments, we conclude that communications cycle and synchronisation are very significant to the localisation performance. When more stable results are required, a longer transmission cycle is the best option. However, when a more reactive system is required, a shorter cycle is more adequate but then the transmissions should be synchronised to reduce collisions and maintain a low standard deviation of the values. This latter case might be more suitable for navigation purposes.

Moreover, the implementation of the sliding window filter, both in window size and data selection, seems to have some impact on the standard deviation of the measurements, albeit small. Consequently, it might be worth exploring some more sophisticated filtering methods.

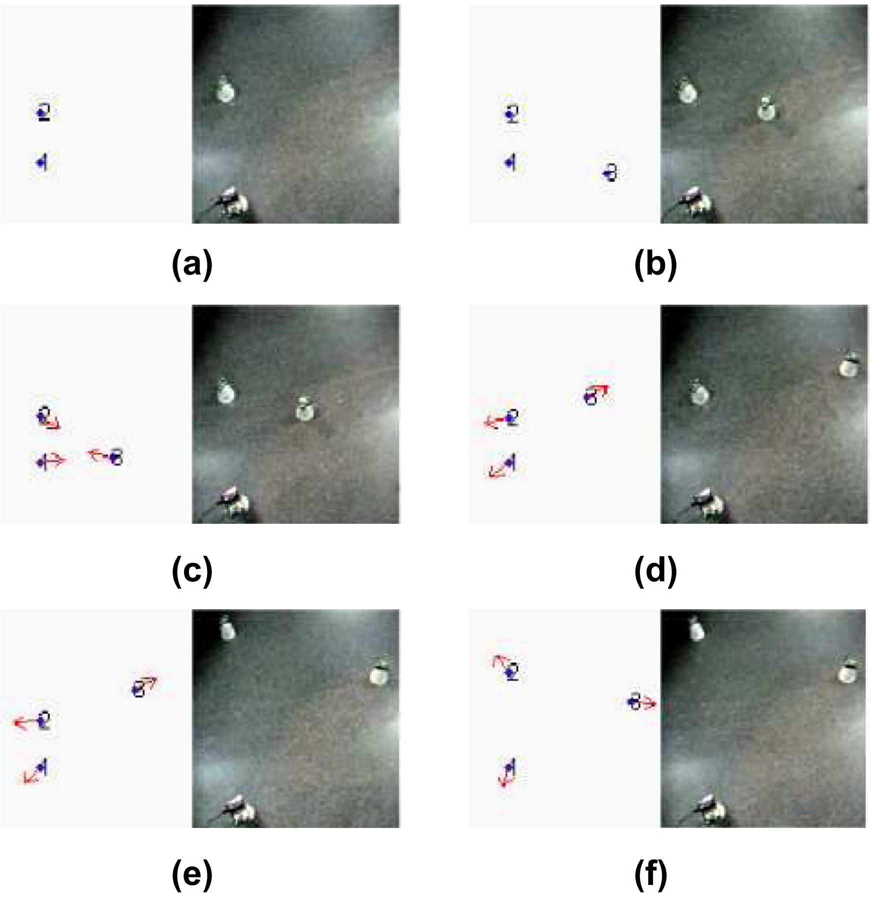

(a) (b) (c) (d) (e) (f)

Fig. 14. Experiment snapshots: Adding and moving nodes in the network (Video available at http://www.youtube.com/watch?v=hfjTys9kv1s). (a) Initial setting – nodes 1 and 2 are static. (b) Node 3 joins the network – the system shows node 3 temporally far from node 1 and 2. (c) Network stabilises – within a couple of periods of computation node 3 approaches nodes 1 and 2. (d) Node 3 moves away – the results show node 3 leaving nodes 1 and 2 (e) Node 2 moves – the positions are not yet refreshed. (f) Network stabilises – After a couple of periods of computation, the estimated relative positions are consistent with the real ones.

Finally, if all nodes are transmitting with equal power and are equipped with similar antennas, the wireless channel is reciprocal, and therefore it is possible to transmit only half of the signal space distance matrix, thus reducing the amount of transmission overhead by approximately half.

6.2. Relative position under mobility

In this section, we show the behaviour of our approach with moving nodes in a few simple situations, as proof of concept. We also show that this localisation system can be used to estimate the signal space velocity of the nodes. The experiments were conducted indoor in a 5 m × 5 m area. The transmission power was set to -15 dBm. The computations were carried out in a PC connected to the MicaZ through the MIB600 board, using Matlab. Note that these experiments were performed in real-time.

6.2.1. Estimating signal space velocity

Fig. 14 shows a series of experiment snapshots comparing the physical positions and computed results. In each sub-figure, the left part shows the computed result from Matlab and the right one depicts the real positions from a camera. The first row in Fig. 14a shows two static nodes, 1 and 2, placed slightly above the floor. We may notice that the signal space velocity vectors are zero, i.e. the signal between nodes 1 and 2 is stable. Then, we added node 3 further away from the two static nodes, as shown in Fig. 14b. However, a couple of periods later, the result shows that they are moving closer to each other, as shown in Fig. 14c, until the relative position estimates with node 3 stabilise. Then, we moved node 3 away from nodes 1 and 2. Within a couple of periods, the system refreshed the relative positions. Then, node 2 was moved upward away from node 1 and node 3 (Fig. 14e). Again, the computed relative positions vary consistently to the true ones.

7. Conclusions

In this paper, we developed a relative localisation algorithm based on the received signal strength indicator of wireless communications as a rough estimate of distance between any pair of communicating nodes. In order to perform the relative localisation, we introduce the so-called signal space distance matrix. Floyd–Warshall algorithm was employed to generate pairwise signal distance for each pair of nodes that were not linked. Multidimensional Scaling was then used to generate relative positions from the pairwise distances, considering a few geometric transformations to stabilise the position estimates. Due to the inherent unreliability of the wireless communications, we studied the impact of several parameters on the network topology estimation, namely different approaches to provide a symmetric distances matrix, the period of the matrix dissemination, the use of synchronisation of the transmissions, and the filtering of the RSSI data. Experimental results with a set of MicaZ nodes showed that the communications period and the use of synchronisation have the highest impact on the network topology. Specifically, we showed that a large communications period yields better precision despite reducing the reactivity of the system. We also showed that shorter periods can be used as long as the transmissions are synchronised, particularly by using the Adaptive-TDMA technique. On the other hand, filtering techniques are shown to have an impact, albeit smaller.

Finally, we showed that in a mobile scenario, using a Kalman Filter together with appropriate coordinates adjustment, it is possible to generate smooth trajectories. Particularly, we showed how the proposed localisation algorithm can be used to estimate signal space velocity and establish a correspondence between orientation in the physical world and in the relative coordinates system. Currently, we are assessing the use of this localisation algorithm in practical formation control situations both in surveillance applications and in robotic soccer in the scope of RoboCup.

Acknowledgements

Luis Oliveira was supported by FCT – Fundacao para a Ciencia e a Tecnologia through the Grant SFRH/BD/74292/2010. Traian E. Abrudan was supported by FCT – Fundacao para a Ciencia e a Tecnologia through the Grant SFRH/BPD/81104/2011.

References

[1] L. Parker, Distributed intelligence: overview of the field and its application in multi-robot systems, Journal of Physical Agents 2 (1) (2008).

[2] H. Li, L. Almeida, Z. Wang, Y. Sun, Relative positions within small teams of mobile units, in: Proceedings of the 3rd International Conference on Mobile Ad-Hoc and Sensor Networks, MSN'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 657–671.

[3] R. Murphy, J. Kravitz, S. Stover, R. Shoureshi, Mobile robots in mine rescue and recovery, Robotics Automation Magazine, IEEE 16 (2) (2009) 91–103.

[4] J. Desai, J. Ostrowski, V. Kumar, Controlling formations of multiple mobile robots, in: Proceedings. 1998 IEEE International Conference on Robotics and Automation, vol. 4, 1998, pp. 2864–2869.

[5] F. Coutinho, J. Barreiros, J. Fonseca, Choosing paths that prevent network partitioning in mobile ad-hoc networks, in: Proceedings, 2004 IEEE International Workshop on Factory Communication Systems, 2004, pp. 65–71.

[6] G. Wang, G. Cao, T. La Porta, W. Zhang, Sensor relocation in mobile sensor networks, in: INFOCOM 2005, 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings IEEE, vol. 4, 2005, pp. 2302–2312.

[7] K. Cheung, H. So, W.-K. Ma, Y. Chan, Least squares algorithms for time-of-arrival-based mobile location, IEEE Transactions on Signal Processing 52 (4) (2004) 1121–1130.

[8] A. Sayed, A. Tarighat, N. Khajehnouri, Network-based wireless location: challenges faced in developing techniques for accurate wireless location information, Signal Processing Magazine, IEEE 22 (4) (2005) 24–40.

[9] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, Z. Sahinoglu, Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks, Signal Processing Magazine, IEEE 22 (4) (2005) 70–84.

[10] N.B. Priyantha, A. Chakraborty, H. Balakrishnan, The cricket location-support system, in: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, MobiCom '00, Springer, New York, NY, USA, 2000, pp. 32–43.

[11] Y. Zhou, C.L. Law, F. Chin, Construction of local anchor map for indoor position measurement system, IEEE Transactions on Instrumentation and Measurement 59 (7) (2010) 1986–1988.

[12] F. Gustafsson, F. Gunnarsson, Positioning using time-difference of arrival measurements, in: Proceedings, (ICASSP '03), 2003 IEEE

International Conference on Acoustics, Speech, and Signal Processing, vol. 6, 2003, pp. 553–556.

[13] D. Niculescu, B. Nath, Ad hoc positioning system (APS) using AOA, in: INFOCOM 2003, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, IEEE Societies, vol. 3, 2003, pp. 1734–1743.

[14] P. Kulakowski, J. Vales-Alonso, E. Egea-Lopez, W. Ludwin, J. Garcia-Haro, Angle-of-arrival localization based on antenna arrays for wireless sensor networks, Computers & Electrical Engineering 36 (6) (2010) 1181–1186.

[15] B. Neuwinger, U. Witkowski, U. Ruckert, Ad-hoc communication and localization system for mobile robots, in: J.-H. Kim, S. Ge, P. Vadakkepat, N. Jesse, A. Al Manum, S. Puthusserypady, K.U. Rückert, J. Sitte, U. Witkowski, R. Nakatsu, T. Braunl, J. Baltes, J. Anderson, C.-C. Wong, I. Verner, D. Ahlgren (Eds.), Advances in Robotics, Lecture Notes in Computer Science, vol. 5744, Springer, Berlin, Heidelberg, 2009, pp. 220–229.

[16] M.B. Jamaa, A. Koubaa, Y. Kayani, Easyloc: RSS-based localization made easy, Procedia Computer Science 10 (0) (2012) 1127–1133 (aNT 2012 and MobiWIS 2012).

[17] T.E. Abrudan, L.M. Paula, J. Barros, J.P.S. Cunha, N.B. Carvalho, Indoor location estimation and tracking in wireless sensor networks using a dual frequency approach, in: IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2011.

[18] G. Blumrosen, B. Hod, T. Anker, D. Dolev, B. Rubinsky, Enhanced calibration technique for RSSI-based ranging in body area networks, Ad Hoc Networks 11 (1) (2012) 555–569.

[19] A. Coluccia, F. Ricciato, A software-defined radio tool for experimenting with RSS measurements in IEEE 802.15.4: implementation and applications, in: 21st International Conference on Computer Communications and Networks (ICCCN), 2012, pp. 1–6.

[20] P. Barsocchi, S. Lenzi, S. Chessa, F. Furfari, Automatic virtual calibration of range-based indoor localization systems, Wireless Communications and Mobile Computing (2011) 1546–1557.

[21] N. Patwari, A.O. Hero I, M. Perkins, N. Correal, R. O'Dea, Relative location estimation in wireless sensor networks, IEEE Transactions on Signal Processing 51 (8) (2003) 2137–2148.

[22] Y. Shang, W. Rumi, Y. Zhang, M. Fromherz, Localization from connectivity in sensor networks, IEEE Transactions on Parallel and Distributed Systems 15 (11) (2004) 961–974.

[23] X. Shen, Z. Wang, P. Jiang, R. Lin, Y. Sun, Connectivity and RSSI based localization scheme for wireless sensor networks, Advances in intelligent computing (2005), 578-587.

[24] M. Rahman, L. Kleeman, Paired measurement localization: a robust approach for wireless localization, IEEE Transactions on Mobile Computing 8 (8) (2009) 1087–1102.

[25] Z.-X. Chen, H.-W. Wei, Q. Wan, S.-F. Ye, W.-L. Yang, A supplement to multidimensional scaling framework for mobile location: a unified view, IEEE Transactions on Signal Processing 57 (5) (2009) 2030–2034.

[26] Y. Shang, W. Ruml, Improved MDS-based localization, in: Twenty-Third AnnualJoint Conference of the IEEE Computer and Communications Societies, INFOCOM 2004, vol. 4, 2004, pp. 2640–2651.

[27] H. Kung, C.-K. Lin, T.-H. Lin, D. Vlah, Localization with snap-inducing shaped residuals (SISR): coping with errors in measurement, in: Proceedings of the 15th Annual International Conference on Mobile Computing and Networking, ACM, 2009, pp. 333–344.

[28] G. Destino, G. De Abreu, Weighing strategy for network localization under scarce ranging information, IEEE Transactions on Wireless Communications 8 (7) (2009) 3668–3678.

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

[30] L. Hu, D. Evans, Localization for mobile sensor networks, in: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, MobiCom '04, ACM, New York, NY, USA, 2004, pp. 45–57.

[31] H. Je, D. Kim, Bayesian multidimensional scaling for multi-robot localization, in: IEEE International Conference on Networking, Sensing and Control, 2008, ICNSC 2008, 2008, pp. 926–931.

[32] I. Borg, P.J.F. Groenen, Modern Multidimensional Scaling: Theory and Applications (Springer Series in Statistics), second ed., Springer, 2005.

[33] T. Facchinetti, G. Buttazzo, L. Almeida, Dynamic resource reservation and connectivity tracking to support real-time communication among mobile units, EURASIP Journal on Wireless Communications Network 4 (5) (2005) 712–730.

[34] G. Welch, G. Bishop, An Introduction to the Kalman Filter.

[35] O.-H. Kwon, H.-J. Song, Localization through map stitching in wireless sensor networks, IEEE Transactions on Parallel and Distributed Systems 19 (1) (2008) 93–105.

[36] O.-H. Kwon, H.-J. Song, S. Park, Anchor-free localization through flip-error-resistant map stitching in wireless sensor network, IEEE Transactions on Parallel and Distributed Systems 21 (11) (2010) 1644–1657.

[37] X. Ji, H. Zha, Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling, in: Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2004, vol. 4, 2004, pp. 2652–2661.

[38] C. Ellis, M. Hazas, A comparison of MDS-MAP and non-linear regression, in: International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2010, pp. 1–6.

[39] A. Amar, Y. Wang, G. Leus, Extending the classical multidimensional scaling algorithm given partial pairwise distance measurements, Signal Processing Letters, IEEE 17 (5) (2010) 473–476.

[40] Chipcon, Cc2420, July 2012. <http://www.ti.com/lit/gpn/cc2420>.

[41] K. Srinivasan, P. Levis, RSSI is under appreciated, in: Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets), 2006.

[42] F. Santos, L. Almeida, P. Pedreiras, L.S. Lopes, T. Facchinetti, An adaptive TDMA protocol for soft real-time wireless communication among mobile autonomous agents, in: Workshop on Architectures for Coopратive Embedded Real-Time Systems, 2004.

[43] F. Santos, L. Almeida, L. Lopes, Self-configuration of an adaptive TDMA wireless communication protocol for teams of mobile robots, in: IEEE International Conference on Emerging Technologies and Factory Automation, 2008, ETFA 2008, 2008, pp. 1197–1204.

Luis Oliveira received the Masters degree in electronics and telecommunications engineering from the University of Aveiro, Portugal, in 2009. He is currently pursuing the PhD degree in the Doctoral Program in Electrical and Computer Engineering, University of Porto, Portugal. His research interests include real-time wireless networks and control architectures for mobile robot teams.

Hongbin Li received his B.E in 2005 and Ph.D in 2010 both in Control Science and Engineering from Zhejiang University. He is currently working at MicroStrategy China Technology Center as a software performance engineer.

Luis Almeida received the Licenciatura degree in electronics and telecommunications engineering and the PhD degree in electrical engineering from the University of Aveiro, Portugal, in 1988 and 1999, respectively. He is currently an associate professor in the Electrical and Computer Engineering Department of the University of Porto, Portugal, and a senior researcher at IEETA and IT Research Units. Formerly, he was a design engineer in a company producing digital telecommunications equipment. His research interests include real-time networks for distributed industrial/embedded systems and control architectures for mobile robots. He is a member of the IEEE Computer Society and the RoboCup Federation.

Traian E. Abrudan received the D.Sc. degree (with honours) from Aalto University, Finland (formerly Helsinki University of Technology) in 2008, and the M.Sc. degree from the Technical University of Cluj-Napoca, Romania in 2000. During 2001–2010, he was a member of SMARAD (SMArt RADios and Wireless Research Unit) which has been selected as Center of Excellence in research by The Academy of Finland. Since September 2010, he has been a postdoctoral researcher at IT, Faculty of Engineering, University of Porto (FEUP), Portugal. His research interests include parameter estimation, numerical optimisation, sensor array signal processing, multicarrier communications and indoor localisation.