



**HAL**  
open science

# A Distributed Method to Localization for Mobile Sensor Networks

Clément Saad, Abderrahim Benslimane, Jean-Claude König

► **To cite this version:**

Clément Saad, Abderrahim Benslimane, Jean-Claude König. A Distributed Method to Localization for Mobile Sensor Networks. WCNC'07: Wireless Communication & Networking Conference, Mar 2007, Hong Kong, China. lirmm-00135595

**HAL Id: lirmm-00135595**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00135595>**

Submitted on 8 Mar 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Distributed Method to Localization for Mobile Sensor Networks

Clément Saad  
LIA/CERI  
339 chemin des Meinajaries,  
BP 1228-84911 Avignon cedex 9  
Email: saad@lirmm.fr

Abderrahim Benslimane  
LIA/CERI  
339 chemin des Meinajaries,  
BP 1228-84911 Avignon cedex 9  
Email: benslimane@lia.univ-avignon.fr

Jean-Claude König  
Université Montpellier 2 - LIRMM  
161 Rue Ada,  
F-34392 Montpellier Cedex 5, France  
Email: konig@lirmm.fr

**Abstract**—Mobile wireless sensors need to know their localizations in many control and monitoring applications. Among all sensors, some know their exact position (i.e., they are equipped with GPS or they are positioned by human intervention). These sensors are called anchors. Some sensors can have different capabilities allowing them to calculate either distances or angles when they receive messages from others nodes. So, they only use anchor positions to obtain an estimated position. However, when sensors are mobile they cannot continuously calculate their position because of the energy constraints. This paper concerns the localization problem in the case where all nodes in the network (anchors and others sensors) are mobile. We propose three techniques following the capabilities of nodes. Thus, each node obtains either an exact position or an approximate position with the knowledge of the maximal error born. Also, we adapt the periods where nodes invoke their localization. Simulation results show the performances of our methods in term of accuracy and determinate the technique the more adapted related to the network configurations.<sup>1</sup>

## I. INTRODUCTION

Ad-hoc wireless sensor networks have been proposed for many applications such as target tracking, intrusion detection, medical applications, climate control, and disaster management. The localization of nodes can be used for routing or others location based services. Sensors are devices, in some cases with scarce resources, which can communicate using wireless communication protocols. Each sensor has a perception radius and if another sensor is in its perception then the two sensors are neighbors. In this network, only some nodes, called **anchors**, know their localizations (i.e., positioned by human intervention or GPS). A maximum number of remaining nodes have to determine their positions based on anchor localizations. The number of anchors has to be as small as possible because sensors equipped with GPS are more expensive and consume more energy. The energy being scarce, sensors have to minimize their computations and, especially, communications. Extensive research efforts have been conducted to resolve the localization problem and many of these propositions assume that sensors are static [1], [2], [3], [4]. This paper deals with the problem of localization in wireless sensor networks when sensors are mobile. There are three scenarios of mobility: sensors and anchors are mobile;

sensors are mobile and anchors are static; sensors are static and anchors are mobile. For the last case, some methods have been proposed [5], [6]. In these methods, mobile anchors can be robots, humans, or other, equipped GPS which are used in order to locate others static sensors. In this paper, we present a new method to resolve the localization problem in the complex scenario where nodes and anchors are mobile. However, this method can be used for the two others cases of mobility. Three schemes are proposed following the capabilities of sensors. Sensors can be equipped with techniques like ToA/TdoA (Time of arrival / Time difference of arrival) or RSSI (Received Signal Strength Indicator) allowing to compute distance between a pair of neighbor sensors. They may also be equipped with AoA (Angle of arrival) technique allowing to compute angle between a pair of neighbor sensors. Finally, sensors may be equipped by none of these techniques. Our method determines an exact position for a sensor when it has at least two anchors in its neighborhood. Otherwise, it gives an approximate position and can compute in this case the generated maximal error. The localization problem with mobile sensors introduces a new problem: in fact, the energy of sensors being weak, each node cannot compute continually its localization in order to maintain accuracy position during its move. Therefore, the question is: *when a node must evoke the calculation of its position?* In [7], authors compare three methods Static Fixed Rate (SFR), Dynamic Velocity Monotonic (DVM) and Mobility Aware Dead Reckoning Driven (MADRD). These methods define periods during which sensors should invoke their localizations. However, the authors assume that when a node invokes its position it obtains an exact localization (e.g. all sensors are equipped with GPS). These methods are explained in section II. However, when only a small number of sensors are anchors, the problem is not addressed. In this paper, we consider this case of network. When a node invokes its localization it does not always obtains its exact position: either it obtains an approximate position or it cannot locate itself. To overcome this problem, our method defines the periods when a node has to invoke its location. Finally, through simulations, we analyze performances of our three techniques. The rest of the paper is organized as follows. Section II describes key existing solutions to the localization problem and especially methods presented in [7]. Section III formalizes this

<sup>1</sup>This work is supported by DGA.

problem and introduces notation system. Section IV presents the details of our techniques. Section V presents experimental results evaluating the performance of our method. Section VI concludes the paper and presents future work.

## II. RELATED WORKS

A large number of existing techniques attempts to solve the localization problem. Some detailed surveys are provided in [8], [9]. These methods assume that each node uses a technique in order to calculate distances or angles with its neighbors. The most popular techniques are RSSI (Received Signal Strength Indicator), ToA/TDoA (Time of arrival / Time difference of arrival) and AoA (Angle of arrival). In RSSI, nodes measure the power of the received signals. With the power transmission information, the effective propagation loss can be calculated; theoretical or empirical models are used to translate this loss into distance. In ToA/TDoA, nodes translate directly the propagation time into distance if the signal propagation speed is known; the most basic localization system using ToA techniques is GPS [10]. In AoA, nodes estimate the angle at which signals are received and use simple geometric relationships to calculate their positions. The accuracy of these measurements is closely related to the network environment; thus, the positions computed by the nodes may contain errors. In this paper, we focus on the localization problem with mobile sensors, especially in case where sensors, anchors or not, are mobiles. Sensors can calculate either distances or angles or none of both. To the best of our knowledge, there is no other method, in the open literature, that deals with this case. Some papers have been proposed in the case where anchors are mobiles and others sensors are static. For example, [5], [6] uses robots or humans, which can be considered as anchors, move in the network and help others nodes to obtain their positions. When sensors are mobile, it is not reasonable that each sensor invokes its localization technique in order to locate itself continually, due to constraint of energy. A first work in [7] proposes three methods SFR (Static Fixed Rate), DVM (Dynamic Velocity Monotonic), MADRD (Mobility Aware Dead Reckoning Driven) to determinate periods where a node invokes its localization technique. But, it assumes that a node obtains its exact position when it invokes its localization (e.g. sensors are equipped with GPS). The following sub-sections explain these three methods.

### A. Static Fixed Rate (SFR)

In this method, each sensor invokes periodically its localization technique with a fixed time period  $t_{sfr}$ . Let  $s$  be a sensor. If  $s$  invokes its localization technique at time  $t$  it obtains its position  $(x_t, y_t)$ . In fact,  $s$  considers that its position is  $(x_t, y_t)$  during period between  $t$  and  $t+t_{sfr}$ . This method does not take into account mobility of the sensors. Specifically, if a sensor is moving quickly, the error will be high; if it is moving slowly, the error will be low.

### B. Dynamic Velocity Monotonic (DVM)

In DVM, each sensor adapts its localization as a function of its mobility: the higher the observed velocity, the faster

the node should be localized to maintain the same level of error. Thus when a node positions it computes its velocity by dividing the distance it has moved since the last localization point by the time that elapsed since the localization. Thus, the node can schedule the next localization point at the time when a specified distance will be covered if the node continues with the same velocity. Therefore, localization will be carried out more often as soon as the node is moving fast. Conversely, localization will be carried out less frequently as soon as the node is moving slowly. Similar to SFR, the location referred by the node between two localization points will be one calculated at the previous localization point.

### C. Mobility Aware Dead Reckoning Driven (MADRD)

MADRD is a predictive protocol that computes the mobility pattern of the sensor and uses it to predict future mobility. If the expected difference between the actual mobility and the predicted mobility reaches the error threshold, then localization should be triggered. This differs from DVM where localization must be carried out when the distance from the last localization point is predicted to exceed the error threshold. Therefore, localization can be carried out at very low frequency, if the node is moving predictably. Otherwise, localization will be carried out more often. In the case where the prediction is perfect, node does not carried out localization. However, the predicted mobility pattern will generally be imperfect. Sensors will typically not follow a predictable model; for example, there may be unpredictable changes of directions or pauses that will cause the predicted model to go wrong. For all these reasons it is necessary to continue localization periodically to detect deviations from the predicted model. In this paper contrary to the previous solutions, we consider the case where all sensors are mobile. We propose a new method to locate sensors and to adapt periodicity to invoke the localization procedure in order to obtain high accuracy while reducing energy consumption. We analyze our solutions and compare them to the previous ones and we adapt them in order to take into account positioning error.

## III. PRELIMINARY

In this paper, we focus on mobile sensor network. Moreover, we assume that all the sensors have identical transmission radius  $r$ ; however, it is easy to adapt our method with sensors having different transmission radius. We represent a wireless sensor network as a graph  $G(V, E)$  where  $V$  is the set of  $n$  nodes representing sensors and  $E$  is the set of  $m$  edges representing communication links. If two nodes  $u, v$  are neighbors, then they are linked and the distance between  $u$  and  $v$  is smaller than  $r$ . We assume also that some anchors have a priori knowledge of their own positions with respect to some global coordinate system (GPS) (black nodes in figures). We consider scenarios where nodes and anchors are mobile. For example, in a military context, soldiers can be equipped with sensors and tanks with anchors. Soldiers use tank positions in order to obtain their positions. Finally, we

should take into account functionalities of each sensors: for example, methods like RSSI or ToA/TDoA and AoA described in previous section, estimate distance or angle between pair of neighbors. In the case where nodes can compute the angles with its neighbors, these angles are calculated related to one direction (north, south, east, west, obtained with compass) and each node uses the same axe. We introduce a notation system in order to take into account all these informations:  $\langle \{M, S\}, \{M, S\}, \{\emptyset, dist, angle\} \rangle$ , the first (resp. second) field defines if nodes (resp. anchors) are mobile or static ( $M$  for mobile,  $S$  for static). The last field determinates the capability of sensor. If a sensor can calculate angles (resp. distances), the value of the last field is assigned to  $angle$  (resp.  $dist$ ). Otherwise, the value is assigned to  $\emptyset$ . In this paper, we focus on configurations  $\langle M, M, \{\emptyset, dist, angle\} \rangle$ .

#### IV. LOCALIZATION TECHNIQUES

Our three techniques take into account capabilities of sensors. The algorithms have to be simple and very quickly due to mobility. Therefore, nodes which do not know their positions, use informations provided by anchors from 1 to  $k$  hops in order to calculate their positions. In our method,  $k = 2$  due to our node mobility model explained in section V. However, it is easy to extend our method with  $k > 2$ .

##### A. Configuration $\langle M, M, \emptyset \rangle$

In this configuration, nodes cannot compute distances or angles when an anchor sends a message. When a node  $X$  wants to know its position, it broadcasts message  $\langle Node_{ask} \rangle$ . Anchors in its neighborhood (anchors at 1 hop) broadcast a message  $\langle Anchor_{ask} \rangle$  in order to know if there are anchors in their neighborhood (anchors at 2 hops from  $X$ ). Finally, each anchor belonging to  $X$ 's neighborhood replies with a message  $\langle (x, y), [(x_1, y_1), \dots, (x_n, y_n)] \rangle$  where  $(x, y)$  represents the coordinates of the anchor sender, and  $[(x_1, y_1), \dots, (x_n, y_n)]$  represents the table of the coordinates of  $n$  anchors belonging to the anchor's neighborhood (i.e., anchors at 2 hops from  $X$ ). Thus,  $X$  obtains two sets  $N_1$  and  $N_2$  representing respectively the set of anchors at 1 hop and 2 hops. Therefore,  $X$  knows that it is neighbor with all anchors belonging to  $N_1$  but not with anchors belonging to  $N_2$ . We assume that each node has the same transmission radius  $r$ . Therefore,  $X$  defines a zone to which it belongs. For example, in figure 1,  $X$  is neighbor with anchors  $A$  and  $B$  ( $A, B \in N_1$ ) but not with  $C$  ( $C \in N_2$ ). So,  $X$  is inside the circles of centers respectively  $A$  and  $B$  and having a radius equal to  $r$ , but outside circle centered in  $C$ .  $X$  knows that it belongs to the zone (represented in strong lines) and computes the center of gravity of this zone and obtains an approximate position  $X'$ . As application, each node represents its one and two hops neighborhood by a grid partitioned in small square boxes. The length of each square side is equal to  $0.1r$ . Each box is initialized by zero. Each square inside the circle of center an anchor (i.e., the position of anchor) belonging to  $N_1$  is incremented by 1. Conversely, each box inside the circle of center an anchor belonging to  $N_2$  is decremented by 1.

Finally, zone containing the node is defined by the boxes with the maximum value. Figure 2 represents the application of the network of figure 1. The intersection zone is defined by boxes with maximal value equals to 2.

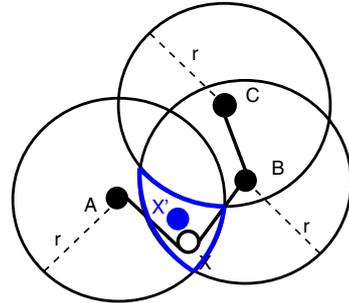


Fig. 1. Approximation principle

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	-1	-1	-1	-1	-1	0	0	0
0	0	0	0	-1	-1	-1	-1	-1	-1	0	0
0	0	1	0	0	0	0	0	0	0	-1	0
0	1	1	0	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	0	0	0	1	0
1	1	1	1	2	2	2	0	1	1	0	0
1	1	1	1	2	2	2	1	1	1	0	0
0	1	1	1	2	2	2	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Fig. 2. Local network representation

Moreover, in this method, a node knows if its estimated position is close to its real position. Let  $\epsilon$  be the distance between the computed center of gravity and the furthest point of the resulting zone. If we consider the distance ( $d_{err}$ ) between the estimated position of a node and its real position representing the estimation error, then the node knows that  $d_{err} \leq \epsilon$ . Therefore, if the value of  $\epsilon$  is small, the node knows that its estimated position is close to its exact position. Conversely, if  $\epsilon$  is high, the node does not know if its estimated position is close to its exact position. Section IV-D explains modification of DVM and MADRD in order to take into account  $\epsilon$ .

##### B. Configuration $\langle M, M, dist \rangle$

In this configuration, when a node receives a message from an anchor, it can compute the distance between itself and the anchor. First, when a node has at least three anchors in its neighborhood, it computes its position with multilateration. Second, when a node has exactly two anchors in its neighborhood, it uses the rule represented in figure 3: when  $X$  receives positions of  $B$ ,  $C$  and  $D$  it deduces that it can be at two positions  $A$  and  $A'$ . Position  $A$  (resp.  $A'$ ) corresponds to the intersection point of the circle centered in  $B$  (resp. in  $C$ ) and with radius equal to  $d_{XB}$  (resp.  $d_{XC}$ ) (distances calculated



and moves toward it. Our simulations use the BonnMotion tool [12] to generate the various scenarios of mobility where velocity and trajectory deviation of nodes vary. Each scenario runs during 90 seconds. *Simulation model:* In our simulations, all messages are delivered. For easier comparison between different scenarios, range errors as well as estimations of position errors are normalized to the radio range. This technique is classical in the literature and allows comparisons with others methods. For example, 50% of position error means a distance equal to half of the radio range between the real and estimated positions. Angle errors are normalized to  $\pi$ . In our scenarios, we use 120 nodes in a square of  $300 \times 300$ . The transmission range of nodes is equal to 20. Among nodes, we randomly select  $\alpha$  anchors with  $\alpha \in \{20, 40, 60, 80, 100\}$  representing a density of anchors in the square from 0.28 to 1.41. Also, we consider measure errors of 0%, 5% and 10% respectively. *Analyse:* In our method, it is possible that a node does not obtain an estimated position when it does not contain anchors in its neighborhood. This case depends on the anchors density. Therefore, if our simulations consider only the position average error rate of sensors, performances of our three techniques would not be shown due to this case. As a consequence, our results focus on the time during which a node is located with a position error lower than 50%. After this time, nodes are considered that they are badly positioned. For our analysis, we perform 1000 tests. For each scenario, we take into account the mean and we represent on graphs the confidence interval. Here, there is 95% of chance that the real value belong to this interval.

*a) Without measure errors:* In this section, we consider the ideal case where measure errors are equal to 0%. The first column of figure 6 shows simulations with SFR, DVM and MADRD. Each one contains three curves which represent respectively the performances of our method without measurement capability (corresponding to  $\langle M, M, \emptyset \rangle$ , called  $x_0$ ), with distance measurement capability (corresponding to  $\langle M, M, dist \rangle$ , called  $x_{dist}$ ) and with angle measurement capability (corresponding to  $\langle M, M, angle \rangle$ , called  $x_{angle}$ ) where  $x$  represents either SFR or DVM or MADRD. These curves represent the time during which a node is located with a position error lower than 50%. For example, in figures 6abc, when the network contains 60 anchors, a node is located with an error lower than 50% during: 18.26s when the node has no capability to calculate neither distances nor angles, 41.01s when the node can calculate distances and 43.18s when the node can calculate angles. Without surprise, accuracy of positions is based on the capability of nodes to calculate distances or angles. For configuration  $\langle M, M, \emptyset \rangle$  (figure 6a), DVM provides better results than SFR and MADRD. It is clear that MADRD provides bad results because it is based on the accuracy of positions and, in this configuration, each node obtains an approximate position. However, as one can expect, in configurations  $\langle M, M, dist \rangle$  (figure 6b) and  $\langle M, M, angle \rangle$  (figure 6c), MADRD provides better results than DVM and then SFR.

*b) Measure errors equal to 5%:* In this section, we introduce measure errors equal to 5%. Figure 6e shows that results obtained when nodes can calculate distances, is not influenced too much by measure errors in SFR, DVM and MADRD. Conversely, results obtained when nodes use angles in figure 6f, are highly influenced by measure errors, related to results shown in figures 6c. More precisely, MADRD uses only accurate previous positions in order to assign position of each sensor. So, in configuration  $\langle M, M, angle \rangle$  it is very influenced by measure errors and its performances are strongly affected. In SFR and DVM, until 60 anchors in the network, configurations using angles are better than configuration using distances and beyond 60 it is the reverse. In fact, when there are small number of anchors in the network, nodes have not often three neighbors. Therefore, a configuration using angles remains better than a configuration using distances. However, beyond 60, it is the reverse because distances are not influenced too much by measure errors. With measure errors equal to 10% this case happens between 30 and 40 anchors in the network. To conclude, graphs in figures 6ef show that in the configuration  $\langle M, M, dist \rangle$ , DVM provides better results than MADRD and then SFR and in the configuration  $\langle M, M, angle \rangle$ , DVM provides better results than SFR and then MADRD.

*c) Conclusions of simulations:* These simulations show the performances of our three methods and show how to adapt SFR, DVM and MADRD, related to the network environment and nodes capabilities in order to provide good results. We note the impact of measure errors in MADRD since it is efficient only if it uses accurate positions. MADRD provides good results in a network environment without measure errors, but when we introduce errors, DVM is the best. Finally, phenomena seen in an environment with measure errors equal to 5%, are confirmed by our simulations with measure errors equal to 10%.

## VI. CONCLUSIONS

This paper proposes three methods for the localization problem when anchors and others sensors are mobile. These methods take into account capabilities of nodes: nodes which can calculate either distances or angles with their neighbors or none of both. Moreover, in order to answer to question *when a node should invoke its position?* related to network environment and capabilities of nodes, we adapted techniques SFR, DVM, MADRD, proposed in [7]. Our simulations show the performances of our methods and determinate the technique the more adapted related to the network configurations.

## REFERENCES

- [1] Dragos Niculescu and Badri Nath. Ad hoc positioning system (aps). *in Proceedings of GLOBECOM, San Antonio*, November 2001.
- [2] Dragos Niculescu and Badri Nath. Ad hoc positioning system (aps) using aoa. *IEEE INFOCOM 2003, San Francisco, CA*, 2003.
- [3] A. Savvides, H. Park, and M. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. *First ACM International Workshop on Wireless Sensor Networks and Application (WSNA), Atlanta, GA*, pp. 112-121, 2002.

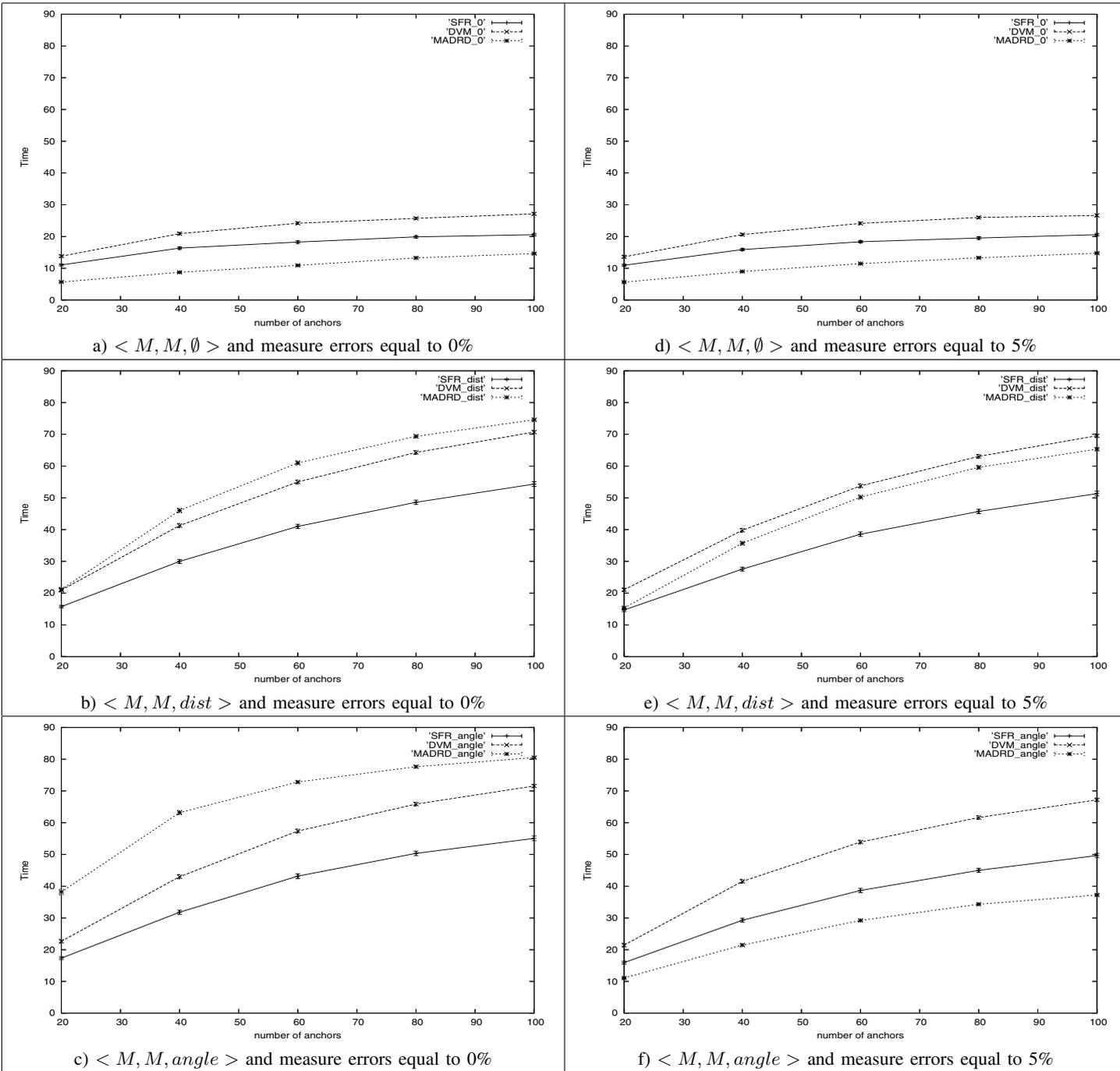


Fig. 6. Performances of SFR, DVM and MADRD in configurations  $\langle M, M, \{\emptyset, dist, angle\} \rangle$  without measure errors (a,b,c) and with errors equal to 5% (d,e,f)

- [4] Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek Abdelzaher. Range-free localization and its impact on large scale sensor networks. *IEEE Personal Communications Magazine*, 2005.
- [5] Nissanka B. Priyantha, Hari Balakrishnan, Erik D. Demaine, and Seth Teller. Mobile-assisted localization in wireless sensor networks. *IEEE Infocom*, March 2005.
- [6] Nirupama Bulusu, John Heidemann, and Deborah Estrin. Adaptive beacon placement. *The Twenty First International Conference on Distributed Computing Systems, Phoenix*, 2001.
- [7] Sameer Tilak, Vinay Kolar, Nael B. Abu-Ghazaleh, and Kyoung-Don Kang. Dynamic localization protocols for mobile sensor networks. *IEEE IWSEASN 2005*, 2005.
- [8] Jeffrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *IEEE Computer* 34(8) pp. 57-66, August 2001.
- [9] K. Langendoen and N. Reijers. Distributed localization in wireless networks : A quantitative comparison. *Computer Networks*, no 43, pp. 500-518, 2003.
- [10] B. Parkinson et al. Global positioning system: Theory and application. *Progress in Astronautics and Aeronautics*, volume I, 1996.
- [11] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trend and Applications*, 2002.
- [12] BonnMotion. <http://web.informatik.uni-bonn.de/iv/mitarbeiter/dewaal/bonnmotion/>.