A Localization Approach in a Distributed Multi-Agent Environment

Muhammad Saim
mnaseem0001@stu.kau.edu.sa

Muhammad Moinuddin
mmansari@kau.edu.sa

Khalid Munawar
kmunawar@kau.edu.sa

Ubaid M. Al Saggaf
usaggaf@kau.edu.sa

Center of Excellence in Intelligent Engineering Systems (CEIES) and Department of Electrical and Computer Engineering,
King Abdulaziz University,
Jeddah, Kingdom of Saudi Arabia.

***Abstract*— Agents in a multi-agent system need to localize themselves and know the positions of the other agents for navigation and good coordination. In the outdoor environment, GPS can solve this problem; however indoors and in the GPS-denied situations, alternate arrangements are necessary. Most of such methods use landmarks or vision-based solutions, and are quite complex and expensive. The objective of this research is to design a simple, yet effective, sensing system for approximate localization of all the agents (or at least those in close vicinity) in a multi-agent environment. The proposed technique uses optical transmitters and receivers available with each agent. An agent transmits a signal to prompt the other agents about the "region" in which it is present. All other agents detect this signal and mark the regions (boundaries) where this agent is detected. When all other agents have marked the boundaries for the transmitting agent, the information gathered is exchanged and intersection region is calculated. The final subset region gives the approximate location of the agent that had transmitted the signal. MATLAB® simulations show the effectiveness of the proposed localization approach. The proposed system is scalable and does not require landmarks; more precise localization is possible depending on the requirements.**

Keywords*— *multi-agent systems; localization; distributed environment; coordination.

I. Introduction

Multi-agent systems have gained a lot of popularity in recent years due to advancement in different technologies, especially communication technologies. Although the concept is not new at all as most of the biological systems work in this fashion, yet its implementation in robotic systems has been a challenge due to different constraints. The main idea is to use the autonomous coordinating agents to solve a, generally, common problem which is either difficult or impossible for individual agents to achieve. However, for effective coordination and successful problem solving, efficient and correct localization is of prime importance. Localization is a process of finding out an agent's position in the whole environment. It is also needed for agents' navigation, collision/obstacle avoidance, and carrying out some specific tasks needing coordination.

Localization in a distributed multi-agent environment is of crucial importance; some of the challenges are to materialize a localization system that is low-cost, less complex but efficient. Furthermore, the agents' localization should be precise enough to avoid collisions and achieve effective co-ordination to carry out the desired tasks. Number of agents can vary; as the number of agents grows, more precise localization is needed to avoid collisions.

Global Positioning System (GPS) is a highly effective solution for global localization in outdoor environments; it can localize an agent quite precisely; yet such localization is not good enough for cooperative task execution. For the indoor environments and places with limited GPS accessibility, however, GPS is helpless [1]. Other positioning methods, based on different sensors and strategies, are needed to achieve localization in such situations. There are various methods and strategies devised; some of them are based on visual information [2] and some others work around dead-reckoning [3] etc. In clean indoor environments with distinct features (like a laboratory or structured environment), it is common to achieve the vision-based localization using fixed camera(s) mounted in the environment through image processing techniques. In tunnels, however, this is not a realistic approach because of vast expanse, poor lighting conditions and smoke. Dead-reckoning is an effective method to achieve high-precision localization. It can be based on sensors directly mounted on the wheels to measure the travelled distance with respect to some reference point, or high-precision inertial sensors mounted anywhere (generally the center of gravity/moment) on the body of a mobile agent and using efficient inertial navigation algorithms to get a good estimate of its location. A major problem with dead-reckoning-based approaches, however, is that they are open-loop and any un-measurable disturbances will cause positioning errors which may not be compensated. Consequently, the integration of these errors will result in complete loss of localization.

This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University (KAU), under grant No. (18-135-35-HiCi). The authors, therefore, acknowledge technical and financial support of KAU.

Several other localization solutions have been proposed in the literature; most of these methods depend on the presence of some landmarks; at least three in most of the cases. In a similar such method, however, the localization is achieved by computing the inter-agent distances and the angles subtended at each agent by lines drawn from two landmarks at known positions [4]. To compensate for the landmarks, a co-operative localization scheme is devised using only two landmarks and inter-agent distances [5]. In another approach, the localization problem is solved using the component based approach. It is done by measuring the time difference between the received wireless radio packet and the ultrasonic pulse; both transmitted simultaneously by the sender node/component; this will give the distance information of the transmitting node from the receiver. Now, by combining or fusing the several distance values, an estimate of the position can be obtained [6]. The benefit of using the component based technique is that components can be easily replaced without affecting the system; they can be deployed and tested separately while integration can be done later. Another solution is to look for a beacon from a landmark present within the environment, and by using distance measurements an agent's localization is achieved. The method used is least squared based adaptive algorithm to achieve asymptotic convergence [7]. This method improves the localization accuracy and exhibits better noise filtering. Furthermore, algorithm converges significantly faster than the previous adaptive algorithms applied on the same problem. In another approach, the localization is achieved through triangulation. Agents can measure not only distances but angles from the landmarks. Corresponding landmarks can be found out using the map already available to the agents. Measuring the bearings of landmarks relative to each other, and finally triangulation would lead to the position of a mobile agent [4]; multiple solutions are obtained for possible location of an agent. Although this method is simple enough but does not have a unique definite solution; further processing with some constraints is needed to get to a unique solution.

In multi-agent environment, Simultaneous Localization and Mapping (SLAM) is a progressing approach [8], [9]. This approach is particularly suited to situations where not just the localization but the environment mapping is also an objective. There are various different implementations of SLAM and those based on probabilistic Bayesian estimation are the most reliable ones as they can manage very well with the noisy measurements and uncertainties in the robot location, the map, and data association especially for those based on particle filters [10], [11]. However, almost all SLAM techniques are quite complex and need heavy computational resources.

In another technique, Radio Frequency IDentification (RFID) tags are placed in the environment (generally, floor) to act as landmarks and RFID readers are attached to a vehicle (generally, bottom) to detect these landmarks. This arrangement is used for detecting position, orientation and velocity of a vehicle [12], [13]. However, since a reader antenna can detect several tags within its detecting range, and the vehicle is on the move while the reader is gathering the data from the tags, the position-estimation error is inherent to this method. Therefore, the precise localization is not easy for the RFID-based systems and a lot of post processing is necessary. Putting more tags, while ignoring the economics, may increase the localization accuracy; but it will also increase the time in reading the tags, which will slow down the detection cycle. Furthermore, an additional expensive reader is necessary to recognize the orientation of the vehicle since the orientation cannot be detected by using only one reader [14].

Cooperative localization is a technique in which multiple vehicles localize themselves by taking advantage of information sharing between them. The process involves location estimation through various estimation tools like extended Kalman filter [15] and particle filters [16] etc. These techniques are centralized and quite difficult to implement in scenarios with large number of agents due to limited communication and computational capabilities. Therefore, a decentralized technique is better suited to this problem; one such approach is localization using covariance intersection filter [17]. However, the complexity and computational cost of all these methods are high enough for economical real-time implementation.

The objective of this research is to devise a methodology for efficient (real-time) and cost-effective localization of the agents in a distributed multi-agent environment. The agents are equipped with communication modules, a processing device and proximity sensors. For this purpose, very simple regional optical proximity sensors and omni-directional transmitters are used. The method is computationally very light and easy to apply to randomly-placed multiple agents in a 2-D plane. However, this method gives only approximate (regional) information about the location of an agent as compared to other more sophisticated methods. The accuracy depends upon how far apart the agents are from each other, how many isolated optical sensors an agent possesses and the total number of agents in a given region. As the agents get closer, the localization also gets better; in this sense it is a practically efficient method. This approach is scalable and its accuracy increases as the number of agents and/or the number of sensors on one agent increase.

Simple computer simulations have demonstrated that the agents can quickly localize the rest of the agents in the whole system (or at least in close vicinity) without causing any significant computational burden on the system; a low-cost microcontroller can easily handle this task in real time.

II. System Model: Problem Formulation

Agents that are to be localized are known to be present in a defined map with known initial positions; agents are also supposed to be dynamic in the environment. An agent is equipped with an omni-directional transmitter (light source) and multiple detectors directed towards different regions around the agent. The detectors are properly isolated to improve their directivity; the positioning of a detector is fixed with respect to the orientation of the agent. Therefore, activity of a sensor will immediately indicate the regions where other agents are located with respect to the agent itself. Following is the layout of the environment and the outline of the procedure of localization in general terms.

A. Time stamp:

A time slot is assigned to each agent and it is also used as the agent's identity and hence to avoid the communication conflicts. All the agents are assigned unique time stamps. It is assumed that all the agents are in close vicinity and able to receive the transmitted light signals from any other agent. In case of obstacles present in the environment, it is assumed that an agent is visible to at least three other agents at a given time instant. Using the time stamp information, all the agents know about which agent should be transmitting a signal at a given instant. The time stamp also works as the identity of an agent. For example, in a 10-agents system, if a time slot is of 100*msec*, then in a 1*sec* detection cycle, the time slot also identifies the agent number. In this sense, there is no confusion for the agents to recognize which agent should be transmitting a pulse at a given time instant.

B. Detection cycle:

An agent A_i keeps track of its clock and when its own assigned time slot comes, it transmits a narrow light pulse that can be seen by all the other agents; the other agents are in reception mode at this time. Because of synchronization, the other agents already know that this is the turn of agent A_i to transmit and, therefore, they expect a transmission from the same. All the sensors possessed by another agent A_j are connected to different pins of its microcontroller but can generate a unified interrupt. As soon as the light is detected, an interrupt will trigger the controller and it will save the state of all the receivers. Only one or a maximum of two sensors looking directly towards A_i would have detected the light and hence the conical region (arc) in which A_i lies is known and marked by A_j to be the region in which A_i is located. Similarly, all the other agents will know the regions with respect to their current locations and orientations in which A_i is present and they will update their information about A_i. This process will continue until all the agents have completed one detection cycle.

C. Computation of subset region:

After one detection cycle is completed as explained above, an agent will know the approximate regions where all other visible agents are located with respect to it. Once a cycle completes, the agents will share their data with the other agents. Each agent will then compute its own location in the complete map by fusing the information acquired from the other agents; this will be done by using simple methods like computing the subset region, points of intersection, triangulation etc. By fusing the information acquired by multiple agents for agent A_i and using history (multiple detection/localization cycles or iterations), the localization accuracy can be significantly improved. The detected subset region will reduce as the agents come closer to each other and that is, in fact, required for collision avoidance during coordination etc. The resulting localization map will then be uploaded to each agent for further utilization in execution of different tasks and collision avoidance etc. The process will start over again.

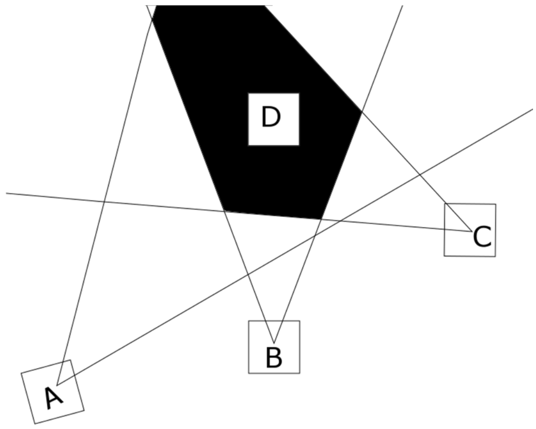

Fig. 1. Subset region of D as marked by A, B and C

As an alternative, the agents will relay their acquired data to a central computer that will compute the current location map and relay the information back to the agents. At any given instant, only one agent can transmit while all the others are in receiving mode. After one complete detection cycle, the agents relay their acquired data to central computer (for localization map computation/update) and receive back the new situation along with the task commands etc.

Subset region is the required localization region of interest and the approximate location of an agent. Fig. 1 shows one example subset region along with the actual location of an agent in that region. There could be multiple ways a subset region is computed depending on the marked boundaries used for the agent detection. If marked boundaries are circular then circular intersection method can be used which is well known in the literature. In case the boundaries are triangular/conical, line intersection method can be used for computing intersection points and then to find the final subset region. If the marked boundaries are rectangular then rectangular intersection method can be used to find the required subset region and this is the approach used in the example simulations given in this paper.

III. RESEARCH METHODOLOGY

In the proposed localization approach, all the agents have precisely synchronized local time stamp generators (clocks). Each agent is assigned a specific dedicated time slot and that also serves as its identity/address. The time slots are frozen; which means that if an agent fails to transmit in its assigned time slot, the communication/processing loop will not stop or corrupted. An agent possesses an omni-directional (in-plane) light source to serve as its transmitter; multiple LEDs installed in different directions can also do the job. This transmitter is to be placed at a point on the agent where it is visible to all other agents or at least to three agents if obstacles are present in the environment. The agent also possesses a set of mutually isolated and properly referenced/addressed light sensors to serve as its receivers/detectors. The accuracy of localization is directly linked to the number of these receivers; each receiver covers an arc (region) around the agent and the number of receivers will define the width of this arc. The localization scheme works as follows.

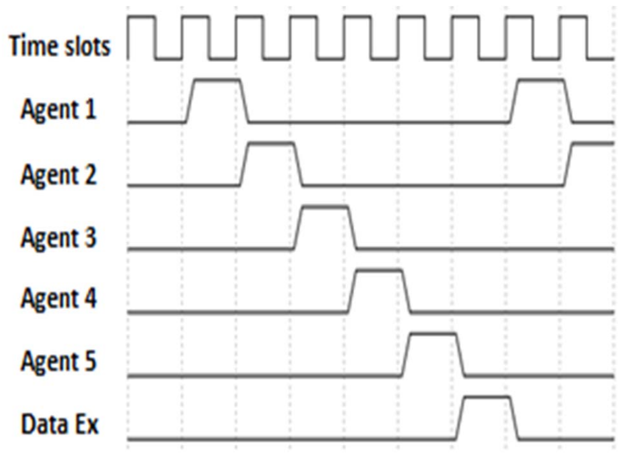

Fig. 2. Timing diagram of different agents (all synchronized clocks)

A system with five randomly positioned agents is considered; the initial positions of the agents are known in a map. The timing diagram is shown in Fig. 2. The ON time represents the time slot when an agent transmits its light pulse while in OFF time it is in receiving mode to detect the other agents. After one complete detection cycle, the agents will share their data among each other or communicate to a central computer as explained above.

For the sake of explanation, it is assumed that an agent has four photo detectors, each one covering one quadrant. Therefore, an agent can be recognized in one quadrant or two depending on its location. As soon as an agent is detected, the detecting agent marks a rectangular region starting from its own last known location and extending to the edges of the working boundaries in that quadrant. This process is shown in Fig. 3. Agent 5 has detected Agent 1 in the 4th quadrant of its current orientation. The marked region extends to the limits of the working boundary. Although an agent can have any arbitrary orientation, for the sake of explanation, however, it is assumed here that all the agents are orientated along the principal axes only. When all the agents have completed their transmission and detection cycles, the gathered information is transmitted to the central computer for the actual localizations of the agents. Once computed, the new map is transmitted back to all the agents and each agent knows the position of all other agents with respect to itself (or in the reference coordinate system). This communication with the central computer is done in the "Data Exchange" cycle shown in Fig. 2. In practical scenario, this communication will be done using the dedicated wireless radio transceivers or the agents may be working in an IP network. Along with the new map, the central computer will also communicate the new tasks or updated commands to be executed. In all this process, for five agents, less than a second is consumed. However, if the number of agents gets larger, this time could increase.

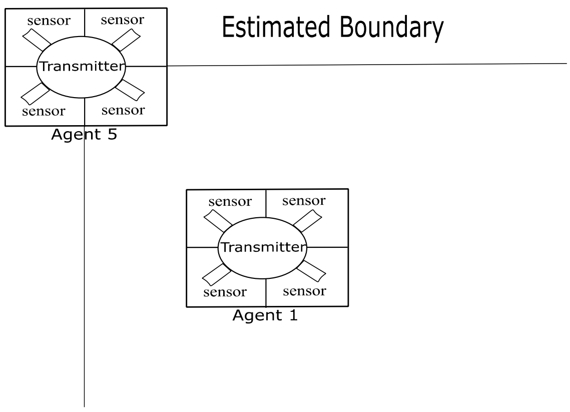

Fig. 3. Detection region of Agent 1 by Agent 5 in the operating field

To find out the subset region (rectangular) for an agent as seen by all other agents, the intersection of all the rectangular detection regions is calculated. The intersection of rectangular regions is well known in the literature.

If four coordinate variables are defined as:

$left$: x – Coordinate of upper left corner
top : y – Coordinate of the upper left corner
$right$: x – Coordinate of lower right corner
$bottom$: y – Coordinate of lower right corner,

and the regions are defined as:

R_i : Subset region in which the agent $\boldsymbol{A_i}$ is located,
R_{ji} : Region in which agent $\boldsymbol{A_j}$ has detected $\boldsymbol{A_i}$, and
R_{ki}: Region in which agent $\boldsymbol{A_k}$ has detected $\boldsymbol{A_i}$,

then the coordinates of the (subset) region R_i are calculated as:

$$R_{i,left} = \max\left(R_{ji,left}, R_{ki,left}\right) \tag{1}$$

$$R_{i,right} = \min\left(R_{ji,right}, R_{ki,right}\right) \tag{2}$$

$$R_{i,top} = \max\left(R_{ji,top}, R_{ki,top}\right) \tag{3}$$

$$R_{i,bottom} = \min\left(R_{ji,bottom}, R_{ki,bottom}\right) \tag{4}$$

This process is illustrated in Fig. 4.

This subset region R_i is the common region detected using the information from any two agents about agent $\boldsymbol{A_i}$. This will be treated and updated subsequently with the region R_{ki} obtained from another agent $\boldsymbol{A_k}$ about $\boldsymbol{A_i}$. This process will continue until $k = j$ and R_i is the subset region of all the regions in which the other agents have detected $\boldsymbol{A_i}$. This will give the smallest region in which the agent is actually located. The process of finding out subset location region for Agent 1 is illustrated in Fig. 5. Five symbols circle, plus, diamond,

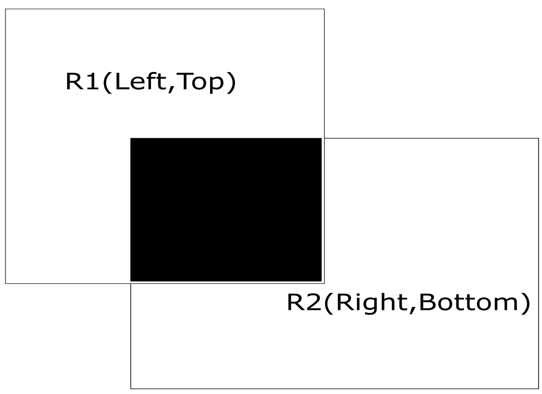

Fig. 4. Computation of the subset region.

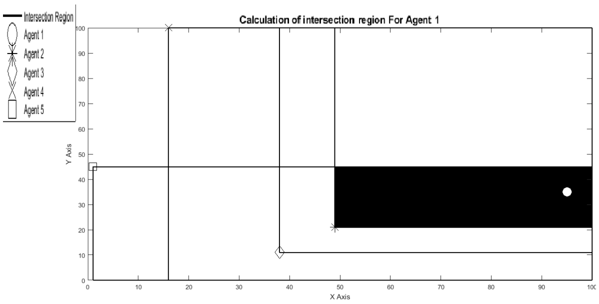

Fig. 5. Final detected region for Agent 1 and its actual position

cross and square represent the actual positions of the agents. For Agent 1 (circle), the other four agents have marked the regions in which they detected Agent 1; marked regions extend to the working boundaries. Intersection of all of these regions is calculated. Finally the intersection region in which Agent 1 is actually located is highlighted in black; this region.

It is clear that this is a scalable solution; increasing the number of agents and/or the sensors an agent possesses, the accuracy of detection will greatly improve. However, for practical purposes, this information is enough. As two agents will come closer, the detection resolution will also increase and hence it will be possible to use that information for collision avoidance etc.

IV. Simulation Results and Discussions

The simulation results of the proposed detection process for all the five agents in the example scenario are presented in Fig. 6. The symbols square, cross, plus, star and circle represent the actual positions of the five agents. Rectangles drawn using symbols circle, plus, cross, triangle and dot represent the estimated region of the agents. It is clear that all the agents could be localized correctly by determining the intersection regions from corresponding estimated rectangular regions. As the regions extend to the working boundaries, they look quite large. However, this will not affect the navigation and collision avoidance because no other agents are present in an extended region. This localization scheme is simple yet effective in correctly identifying the approximate location of all the agents. When agents come closer to each other, the regions become smaller and will allow more precise navigation, coordination and collision avoidance.

The best part about this method is that the detection accuracy is directly proportional to number of agents; therefore this scheme is completely scalable and extendable to a large number of agents. However, as more time slots will be needed for increased number of agents, the detection cycle will be longer, communication of data and then estimation of subset regions will involve more computations, hence the updated estimates will be available at lower frequency. Despite that, as the computations are indeed trivial, this scheme can be easily implemented on simple embedded computers for the distributed multi-agent scenario.

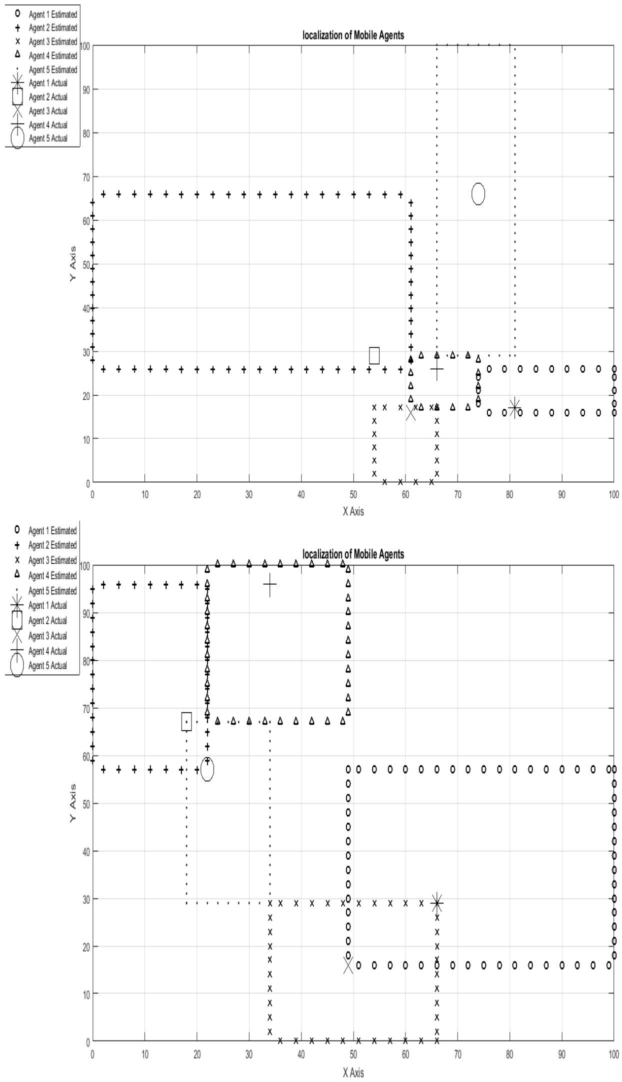

Fig. 6. Simulation results for five agents localization in two different scenarios

V. Conclusions

In most of the localization schemes for GPS-denied situations, the information of the known landmarks is used. In this paper, a simple localization scheme that does not require designated landmarks is presented. In this scheme, each agent transmits an omni-directional light signal in its designated time slot, which also acts as its identity, and all the other agents detect this signal with their multiple regional detectors to mark the regions in which they have sensed the transmitting agent. Subset of all these regions detected for an agent will generate a smaller region in which the agent is actually located. This region will shrink enough for utilization in practical coordination when the agents come closer to each other. The presented scheme is easier to implement and computationally inexpensive allowing cheaper embedded electronics to easily handle it.

The simulations with only four regional detectors on each agent have shown the feasibility and utility of this approach for multi-agent localization and coordination. The approach is scalable and the localization accuracy can be easily improved by using more sensors/detectors and/or increasing the number of agents. The only limitation here is the range and sensitivity of the sensors that can limit the region of operation in a multi-agent system spread over a wide area. The presented scheme is scalable and can efficiently work in an arena with obstacles and occlusions under the stated assumptions.

REFERENCES

[1] A. El-Rabbany, "*Introduction to GPS: The global positioning system,*" 2nd ed., Boston, MA: Artech House Publishers, 2006.

[2] J. Spletzer, A. K. Das, R. Fierro, C. J. Taylor, V. Kumar and J. P. Ostrowski, "Cooperative localization and control for multi-robot manipulation," *Proceedings of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems,* Maui, HI, vol.2, pp. 631-636, 2001.

[3] H. J. von der Hardt, D. Wolf and R. Husson, "The dead reckoning localization system of the wheeled mobile robot ROMANE," *Proceedings of 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems,* Washington DC, pp. 603-610, 1996.

[4] I. Shames, B. Fidan, B. D. O. Anderson and H. Hmam, "Self-localization of mobile agents in the plane," *Proceedings of 3rd ISWPC International Symposium on Wireless Pervasive Computing,* Santorini, pp. 116-120, 2008.

[5] I. Shames, B. Fidan, B. D. O. Anderson and H. Hmam, "Cooperative self-localization of mobile agents," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 3, pp. 1926-1947, 2011.

[6] P. Alriksson *et al*., "A component-based approach to localization and collision avoidance for mobile multi-agent systems," *Proceedings of 2007 European Control Conference (ECC)*, Kos, pp. 4285-4292, 2007.

[7] B. Fidan and A. Çamlıca, "Least-squares based adaptive source localization by mobile agents," *Proceedings of 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, pp. 1286-1291, 2012.

[8] J. L. Blanco, J. A. FernÁndez-Madrigal and J. GonzÁlez, "Toward a unified Bayesian approach to hybrid metric-topological SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 259-270, 2008.

[9] Nosan Kwak, In-Kyu Kim, Heon-Cheol Lee and Beom-Hee Lee, "Adaptive prior boosting technique for the efficient sample size in fastSLAM," *Proceedings of 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, pp. 630-635, 2007.

[10] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," *Proceedings of AAAI National Conference on Artificial Intelligence*, pp. 593–598, 2002.

[11] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229-241, 2001.

[12] J. L. M. Flores, S. S. Srikant, B. Sareen and A. Vagga, "Performance of RFID tags in near and far field," *Proceedings of 2005 IEEE International Conference on Personal Wireless Communications (ICPWC 2005)*, pp. 353-357, 2005.

[13] V. Kulyukin, C. Gharpure, J. Nicholson and S. Pavithran, "RFID in robot-assisted indoor navigation for the visually impaired," *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, vol.2, pp. 1979-1984, 2004.

[14] S. Han, H. Lim and J. Lee, "An efficient localization scheme for a differential-driving mobile robot based on RFID system," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 6, pp. 3362-3369, 2007.

[15] N. Karam, F. Chausse, R. Aufrere and R. Chapuis, "Cooperative multi-vehicle localization," *Proceedings of 2006 IEEE Intelligent Vehicles Symposium*, Tokyo, pp. 564-570, 2006.

[16] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A probabilistic approach to collaborative multi-robot localization," *Autonomous Robots,* vol. 8, no. 3, pp. 325–344, 2000.

[17] P. O. Arambel, C. Rago, and R. K. Mehra, "Covariance intersection algorithm for distributed spacecraft state estimation," *Proceedings of American Control Conference*, pp. 4398-4403, 2001.