

Distributed Robot-assisted Node Localization in Active Sensor Networks

Weihua Sheng, Girma Tewolde
Electrical and Computer Engineering Dept.
Kettering University/Formerly GMI
Flint, MI 48504
Email: wsheng@kettering.edu

Yi Guo
Dept. of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, NJ 07030

Abstract— In this paper we proposed a new sensor network architecture called active sensor network (ASN) where multiple small, sensor network-friendly mobile robots are integrated into a traditional sensor network. Such sensor networks have the following merits: adaptivity, self healing, responsiveness and longer lifetime. The focus of this paper is to develop a distributed sensor node localization with the help of multiple mobile robots. A potential-based robot area partition algorithm and a localization algorithm are developed. We verified our proposed algorithms in simulation.

I. INTRODUCTION

A. Motivation

Recent advancements in wireless communication and micro-electro-mechanical systems (MEMS) have made possible the deployment of wireless sensor networks for many real world applications, such as environmental monitoring, search and rescue, military surveillance, and intelligent transportation [1], [2], [3], etc. In many situations, people do not have the luxury to carefully distribute the sensors. Instead, the sensors are deployed in large quantities very quickly. For example, in environmental monitoring and military surveillance, sensors can be dropped from airplanes. At disaster sites, search and rescue sensor networks are manually deployed by rescue workers in a quick fashion. However, the inherited problems with such traditional wireless sensor networks are: (1) It is very difficult to control the sensor density, coverage and connectivity of the sensor networks. (2) Once deployed, the sensor network can not adapt to a changing environment. (3) The overall life time of the sensor network is limited by the capacity of the batteries carried by the sensors and the network will not be able to carry out the mission if a critical number of sensors deplete their batteries. (4) There lack efficient methods to quickly determine the geographic locations of a large number of sensors while the location information is very important to most of the applications.

In this work, contrary to traditional “open loop”, passive sensor networks, we develop a new sensor network architecture called *active sensor network*, which employs multiple sensor-network-friendly micro service robots (μ ServBots) to implement an actuation mechanism and thus closes the loop. Based on a set of core functions, the micro service robots

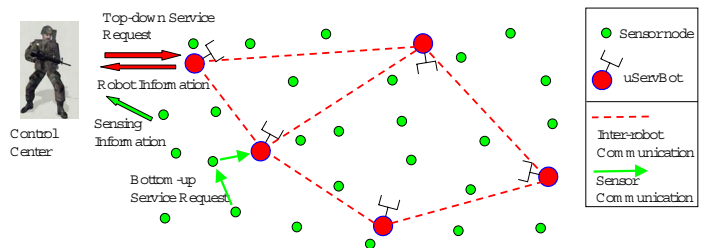


Fig. 1. An active sensor network: closing the loop through actuation.

can provide both logistic and network services. Examples of the logistic services may include (1) sensing coverage control; (2) sensor power supply and (3) sensor calibration, etc. Examples of the network services may include (1) network connectivity, or topology management; (2) hierarchical routing and (3) time synchronization, etc. When these micro service robots are deployed together with a large quantity of sensors, the resulted active sensor network will achieve many desirable merits, such as adaptivity, self healing, responsiveness and longer lifetime. The active sensor network architecture is illustrated in Figure 1, where logistic or network service requests are either initiated by the control center or by the sensors. Here the control center, which can be a soldier, a firefighter, a rescue worker, or simply a computer, is not just an information sink [1] as in many traditional “open loop” sensor networks. Instead, it can actively generate commands to control and manage the underlying sensor network.

B. Related work

Embedding mobile robots in sensor networks has received some attention recently. LarMaca *et al.* [4] proposed a sensor network that uses a robot to carry out the following functions: deploy and calibrate sensors, detect and react to sensor failures. Corke *et al.* [5] used a UAV (an autonomous helicopter) to quickly deploy sensors for large-scale environmental monitoring purpose. They then used the UAV to discover the topology of the deployed sensor network and repair the network to achieve certain connectivity. Bychkovskiy *et al.* proposed a sensor network to investigate the control and actuation in data-centric wireless sensor

networks [6]. However, we have observed the following gaps between the proposed active sensor network and the current research work in sensor networks embedded with robot-driven mobility: (1) The robots used in existing research work are either commercial robots which are too expensive to be employed for real applications, or simple microrobots with very limited capabilities in computation, localization and navigation. To develop an active sensor network for real world applications, sensor-network-oriented robots which can be smoothly integrated into the underlying sensor network, should be developed. (2) In existing work, no systematic models, approaches and methodologies have been developed regarding the control and management of multiple mobile robots in the context of a sensor network. The scalability problem, which certainly needs to be solved in large active sensor networks, has not been addressed yet.

This paper is organized as follows: Section II presents the overall framework of the active sensor network and the hardware platform. Section III introduces the sensor node localization problem and proposes a distributed, multiple robot-based localization algorithm. Simulation results are provided in Section IV and Section V concludes the paper.

II. OVERALL FRAMEWORK

In order to provide the above-mentioned logistic and network services, we identify the following four core functions that the μ ServBots should implement: (1) sensor localization; (2) service set partition; (3) sensor network-assisted inter-robot communication and (4) distributed task allocation. Sensor localization is the process of determining the geographic location of each sensor, which is very important to many sensor network applications. Service set partition aims to divide the sensors into multiple subsets so that each μ ServBot is mainly responsible for one subset. This provides a scalable solution to the maintenance of a large network. Sensor-network-assisted inter-robot communication utilizes the underlying sensor network to provide a backup communication channel when two robots can not directly talk to each other. Distributed task allocation addresses how to distribute the given task among multiple μ ServBots while maximizing the energy and time efficiency. Each high level service relies on one or more of the core functions provided by the μ ServBots.

A. The design of μ ServBots

We find the following capabilities are necessary for the μ ServBots to implement the core functions and the desired logistic and network services: all-terrain mobility; autonomous navigation; sensor transportation; energy harvest/supply; ranging and wireless communication. Figure 2 shows the hierarchical relationship between the services, the core functions and the capabilities of the μ ServBots.

The conceptual sketch of the μ ServBot is illustrated in Figure 3. In order to provide all-terrain mobility, tracks are adopted for μ ServBots. Two DC motors provide the driving force in a differential fashion. To provide autonomous

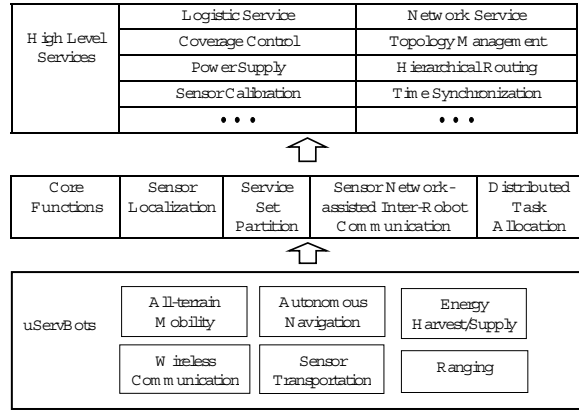


Fig. 2. The high level services, core functions and capabilities of the μ ServBots.

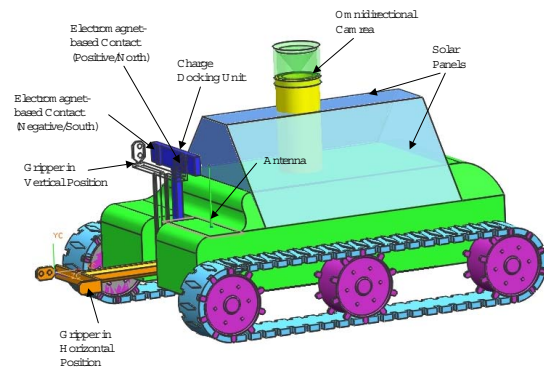


Fig. 3. The conceptual sketch of the μ ServBot.

navigation capability, the following navigation sensors are mounted on the μ ServBot: (i) a miniature omnidirectional camera; (ii) a miniature GPS; (iii) a 3-axis inclinometer and (iv) several IR proximity sensors. The omnidirectional camera provides information about the local environment surrounding the μ ServBot and provide visual guidance to approach and attend the sensors. The GPS and the inclinometer provide location and orientation information crucial to the navigation. To provide sensor transportation capability, each μ ServBot is equipped with a gripper. Two servo motors are used to drive the gripper to open/close, and rotate. Energy harvest can take different forms in different applications. Automatic recharging, vibration energy collection, wind power, etc. are some examples. We use onboard solar panels to collect the power for the μ ServBots. Energy supply to sensors is realized through a novel charge unit on the μ ServBots. To charge a sensor, a μ ServBot first picks up a sensor which is mounted in a specially designed universal adaptor (see next section). The gripper then docks the sensor to the onboard charge docking unit, which is mounted on a T-shape supporting frame and consists of a pair of electromagnet-based charging contacts. The polarity of the electromagnets and the charging contacts can be reversed simultaneously to match the polarity of permanent magnets and power contacts on the sensor adaptor. This feature can

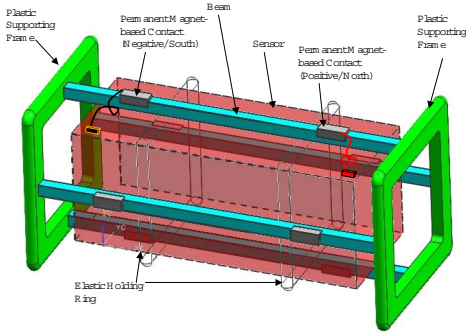


Fig. 4. The universal sensor adaptor. Note: all 8 magnet-based contacts are connected to the power contacts on the sensor.

be implemented by a small touch sensor mounted against the T-shape supporting frame.

B. The design of the universal sensor adaptor

To enable the robot-sensor interaction, a lightweight, low-cost universal sensor adaptor is developed. Figure 4 shows the design sketch of the universal sensor adaptor. This adaptor consists of a supporting frame and 8 permanent-magnet-based charging contacts with two on each of the four beams. The sensor is held in the two rubber rings attached to the four beams. In order to make the sensors easy to be recognized by the μ ServBots, the supporting frame will be painted with certain colors. This universal adaptor serves multiple purposes: (1) facilitating the battery charging; (2) protecting the sensor by absorbing impacts during deployment; and (3) making the sensor easy to identify and grasp. This sensor adaptor can also be equipped with other optional components, such as a tiny acoustic generator/receiver to support accurate ranging capability.

C. The design of μ ServBot control system

As shown in Figure 5, a hierarchical architecture is adopted for the control system. At the top level, an Intel Xscale 400 CPU (PXA255 with 64M RAM and 8M flash memory) is responsible for the high level decision making, motion planning and image processing. It runs Linux operating system. At the bottom level, PIC microprocessors are used to control individual sensing and actuation modules. They communicate with the Xscale 400 main CPU using I^2C bus. There are two communication channels on the μ ServBots. One channel is for direct inter-robot communication, which has higher bandwidth. A Wi-Fi module is adopted and driven by the main CPU. The other channel is for robot-sensor communication, which has lower bandwidth. To facilitate this communication, a sensor interface is designed so that a sensor can be mounted on the μ ServBot through this interface. Zigbee based MicaZ motes from Crossbow Technologies Inc. are used as the sensors. Therefore a μ ServBot is able to talk to the sensors over Zigbee. The sensing module consists of the omnidirectional camera, the GPS, the 3-axis inclinometer and the set of IR proximity sensors. The actuation module consists of

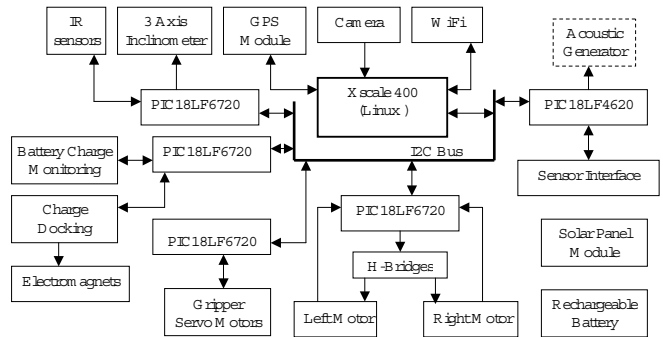


Fig. 5. The control system design of the μ ServBots.

three units: the main motor drive unit, the gripper drive unit and the docking control unit. The main motor drive unit includes a PIC microprocessor, two H-bridges and two optical encoders for speed feedback. The gripper drive unit includes a PIC microprocessor, two servo motors and the associated circuit. The docking control unit includes the relay and electromagnets. The power module consists of two solar panels, a rechargeable Li-Ion battery, and a charging circuit which is responsible for the management, monitoring of the charging of the onboard robot batteries and the sensor batteries.

III. THE DISTRIBUTED SENSOR LOCALIZATION ALGORITHM

An important problem in many sensor network applications is to find out the geographic locations of the sensor nodes. In recent years, several sensor localization methods have been developed for ad hoc wireless sensor networks. Most of the node localization algorithms are based on range measurement, through either time of arrival (TOA) [7], time difference of arrival (TDOA) [8], or received signal strength (RSS) [9]. For example, In the Picoradio project [10] at UC Berkeley, a geolocation scheme for an indoor environment is provided based on RF received signal strength measurements and pre-calculated signal strength maps. The AHLoS (Ad-Hoc Localization System) [11] proposed by Savvides *et. al* enables sensor nodes to discover their locations using a set distributed iterative algorithms. An RF based proximity method was developed by [9], in which the location of a node is given as a centroid generated by counting the beacon signals transmitted by a set of beacons pre-positioned in a mesh pattern. Other methods that do not rely on range measurements were also developed. For example, the count of hops is used as an indication of the distance to the beacon nodes in some applications [7].

In this paper, we develop a μ ServBot-assisted distributed node localization algorithm. Since each μ ServBot is equipped with a positioning module such as the GPS, the μ ServBots can play as mobile beacons in the localization process. The basic idea is as follows. The μ ServBots move around and frequently send out *localization broadcasts*, which are RF messages embedded with senders' current

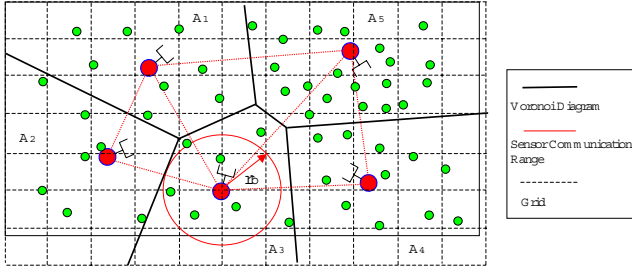


Fig. 6. Locating the sensors with μ ServBots.

location. A sensor receiving localization broadcasts from 3 or more different locations is able to recover its own location through lateration, using the distances calculated from the strength of the received signals [9]. If the μ ServBots are equipped with an acoustic generator and the sensors are equipped with acoustic receivers on the adaptor, more accurate distance measurement can be obtained through Time Difference of Arrival (TDOA) technique.

In order to localize all the sensors, a method is needed to ensure that any sensor in the area of interest is able to hear at least 3 different localization broadcasts (RF message or RF message with acoustic signal) from one or more μ ServBots. To ensure sufficient accuracy, the localization broadcasts should be sent from distinct locations. On the other hand, in order to save power and minimize the localization time, we also want to minimize the total traveling distance of the μ ServBots. To achieve this, the area of interest \mathcal{A} , usually a rectangle defined by its four vertices, is placed with grids as shown in Figure 6. The spacing is set to $\frac{\sqrt{2}}{2}r_b$, where r_b is the sensor RF range, or the minimum of the sensor RF range and the acoustic signal range. Therefore, as long as every grid point is visited by one of the μ ServBots and a localization broadcast is sent out at that grid point, any sensor will receive at least four distinct localization broadcasts. To achieve this, a scheme is needed to coordinate the μ ServBots to visit all the grid points. The proposed cooperative multi-robot sensor localization will be conducted in the following three steps:

(1) μ ServBots disperse and partition the area of interest into subareas that have roughly equal size.

(2) Each μ ServBot visits the grid points in its associated subarea and sends out localization broadcast at each grid point.

(3) Each μ ServBot collects the location information of the sensors in its own subarea.

In the first step, μ ServBots disperse themselves into the area of interest \mathcal{A} . To achieve uniform dispersion, a potential-based algorithm is adopted [12]. The basic idea is to take the boundary of the area of interest \mathcal{A} as obstacles that the μ ServBots should keep away from, and the μ ServBots expel each other using potential forces until a stable deployment is achieved. The detailed algorithm is as follows:

Potential-based deployment algorithm for μ ServBot R_i
 /* starting from the initial position p_i . $\mathbf{r}_{ij} = p_j - p_i$ is the

relative position from μ ServBot R_i to μ ServBot R_j . $\mathbf{r}_{ik} = O_k - p_i$ is the relative position from R_i to obstacle O_k */.

- (1) Exchange location information p_i with other μ ServBots.
- (2) Calculate the the force caused by the obstacles:

$$\mathbf{F}_{oi} = -k_o \sum_k \frac{1}{r_{ik}^2} \cdot \frac{\mathbf{r}_{ik}}{r_{ik}}$$

Calculate the force caused by other robots:

$$\mathbf{F}_{ni} = -k_n \sum_j \frac{1}{r_{ij}^2} \cdot \frac{\mathbf{r}_{ij}}{r_{ij}}$$

Calculate the overall force exerted on robot R_i :

$$\mathbf{F}_i = \mathbf{F}_{oi} + \mathbf{F}_{ni}$$

- (3) If $\mathbf{F}_i \geq \mathbf{F}_{th}$ then change the velocity of R_i according to the following equation:

$$v_i = v_i + (\mathbf{F}_i - \kappa v_i) / m \cdot \Delta T$$

where \mathbf{F}_{th} is a small force threshold. κ is a viscous friction coefficient.

- (4) Update the location of R_i through the following equation:

$$p_i = p_i + v_i \cdot \Delta T$$

- (5) Go to (1).

Here k_o and k_n are force constants for robot-obstacle interaction and inter-robot interaction, respectively. The settings of these constants will determine the contribution of each component of force in the net force applying on the μ ServBot. The viscous friction coefficient κ will help minimize oscillations and ensure that the system will reach steady state as the forces approach zero. ΔT is the time step for each iteration and m is the mass of the μ ServBot.

As proved in [12], due to the existence of the viscous force κv_i , the μ ServBots will eventually converge to an equilibrium point, which is the final deployment. To make the algorithm easy to scale up, each μ ServBot can only interact with its neighbors instead of all the robots, which will maintain the same convergence due to the fact that remote μ ServBots generate very small forces.

Once the μ ServBots are deployed, a Voronoi Diagram [13] D_v is constructed, which partitions \mathcal{A} into subareas $\mathcal{A} = \{A_1, A_2, \dots, A_n\}$. It also gives a partition of the grid points set S_g into $\{S_{g1}, S_{g2}, \dots, S_{gn}\}$. In the second step, μ ServBot R_i will visit the grid points in its associated set S_{gi} . At each grid point, a localization broadcast will be sent out. Due to the regular pattern of the grid points, the path planning is trivial. For example, a zigzag pattern can be used to visit all the grid points. Upon receiving four localization broadcasts, a sensor calculates its location through lateration. In the third step, the location information of all the sensors in a subarea is collected by the associated μ ServBot, which can be done by sending query messages to the corresponding sensors.

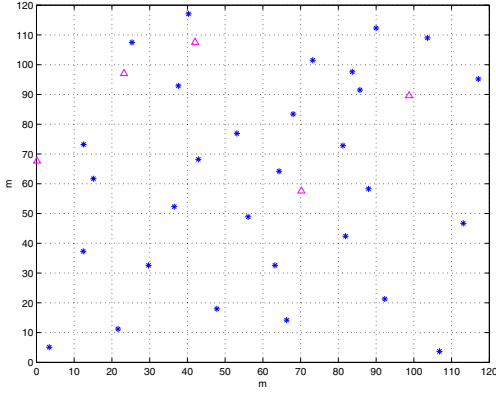


Fig. 7. The random distribution of the μ ServBots and the sensor nodes. The triangles represent the μ ServBots and the stars represent the sensor nodes.

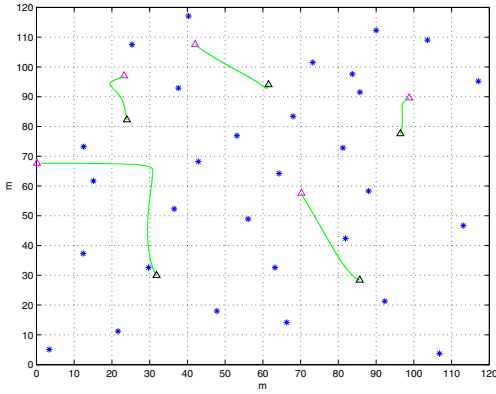


Fig. 8. The trajectories of the μ ServBots.

IV. SIMULATION RESULTS

We conducted simulation of our distributed localization algorithm in C and *Matlab*. A region of $120m \times 120m$ is considered. Five μ ServBots are randomly deployed. The initial random locations of the μ ServBots and the sensor distribution are shown in Fig 7.

First, the potential field-based robot deployment algorithm is applied to compute an optimal location for each μ ServBot to achieve even coverage of the region. The parameters used in the algorithm are as follows:

$$k_o = 48.0, k_n = 80.0, \kappa = 0.5$$

$$\Delta T = 0.1s, m = 1.0kg$$

The paths followed by the μ ServBots during the deployment are shown in Fig 8. Then the Voronoi diagram is constructed to partition the region into 5 approximately equivalent subareas, as shown in Figure 9.

The application of the localization algorithm in each subarea then follows. Each μ ServBot moves in a zigzag fashion along the grid points. The RF range, or the minimum of the sensor RF range and the acoustic signal range is set to be $r_b = 10\sqrt{2}m$. Therefore the grid spacing is $10m$. To emulate real world situations, as the sensors

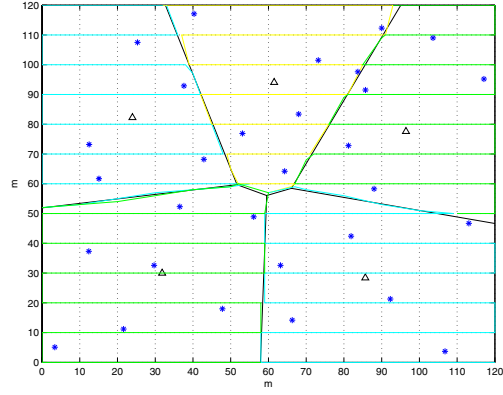


Fig. 9. The partition of the area based on Voronoi diagram.

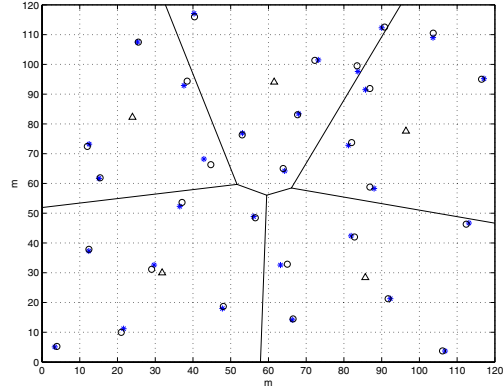


Fig. 10. The localization results: circles represent the estimated sensor locations and the stars represent the actual sensor locations.

collect their data, a Gaussian noise is introduced in the robot location estimation and distance measurement. The Gaussian noises introduced in the robot location estimation and distance measurement have a mean value of $0m$ and a standard deviation of $0.5m$. When a μ ServBot arrives at each grid point it broadcasts its location to its surrounding sensor nodes. Each sensor node then collects at least four localization broadcasts as the μ ServBot reaches within its listening area. Then the location of the sensor nodes is calculated. The complete localization results of the sensor nodes are shown in Fig 10.

We also ran the algorithms on different data sets and recorded the average and maximum localization errors. Two sets of Gaussian noises are used: (1) mean = $0m$ and standard deviation = $0.2m$ (2) mean = $0m$ and standard deviation = $0.5m$. The following tables show the average and maximum localization errors.

Note that the error in the position estimate (in each of the two test cases with different levels of noise) appears to be higher than the level of noise induced in the system. The reason is due to the combined uncertainty of the robot locations and the distance measurements. So when each sensor node is estimating its position the errors in the anchor node positions and the distance measurements between the

TABLE I

TESTING RESULTS (GAUSSIAN NOISE MEAN=0M, STD. DEV.=0.2M)

| Test runs | Average error (m) | Standard deviation (m) |
|-----------|-------------------|------------------------|
| 1 | 0.40 | 0.37 |
| 2 | 0.40 | 0.28 |
| 3 | 0.41 | 0.33 |
| 4 | 0.42 | 0.32 |
| 5 | 0.41 | 0.22 |
| 6 | 0.44 | 0.24 |
| 7 | 0.36 | 0.22 |
| 8 | 0.47 | 0.31 |
| 9 | 0.31 | 0.33 |
| 10 | 0.39 | 0.36 |

TABLE II

TESTING RESULTS (GAUSSIAN NOISE MEAN=0M, STD. DEV.=0.5M)

| Test runs | Average error (m) | Standard deviation (m) |
|-----------|-------------------|------------------------|
| 1 | 1.05 | 0.68 |
| 2 | 1.20 | 0.81 |
| 3 | 1.02 | 0.52 |
| 4 | 1.09 | 0.70 |
| 5 | 1.01 | 0.63 |
| 6 | 1.02 | 0.61 |
| 7 | 1.06 | 0.64 |
| 8 | 1.18 | 0.70 |
| 9 | 1.04 | 0.61 |
| 10 | 1.20 | 0.74 |

sensor nodes and the anchor node as it moves along the grid points causes the error in the position estimate from the lateration step to be higher. But the main observation is that even with this apparently higher level of location error compared to the input noise, the location estimates are good with respect to the system dimensions and the grid size. That is, a mean error of 0.4m with an std of 0.3m on a grid size of 10m x 10m is believed to be a good result for typical sensor network applications. From the above tables we can see that the distributed localization algorithm is effective.

V. CONCLUSIONS

This paper proposes a new sensor network architecture by utilizing multiple micro mobile robots. Such a closed-loop sensor network has many merits that traditional sensor networks do not have. As one of the core functions carried out by the mobile robots, node localization is studied and a distributed, multiple robot-based algorithm is proposed. Simulation results prove that the algorithm is effective and the sensor nodes can be localized with reasonable accuracy. We expect with more accurate self-positioning algorithms implemented on the μ ServBots, such as vision-assisted localization, the sensor node location accuracy can be further improved. Our future research will focus on the development of various algorithms for the other core functions that the μ ServBots should have.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Commun. Mag.*, 40:102–114, 2002.

[2] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of Wireless Sensor Network and Applications*, 2002.

[3] S. N. Simic and S. Sastry. Distributed environmental monitoring using random sensor networks. In *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks*, pages 582–592, 2003.

[4] A. Larmaca, W. Brunette, D. Koizumi, M. Lease, S. B. Sigurdsson, K. Sikorski, D. Fox, and G. Borriello. Making sensor networks practical with robots. In *International Conference on Pervasive Computing*, 2002.

[5] P. Corke, S. Hrahar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme. Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle. In *IEEE International Conference on Robotics and Automation*, 2004.

[6] V. Bychkovskiy, T. Schoellhammer, and D. Estrin. Control and actuation in data-centric wireless sensor networks. In *INTELLIGENT DISTRIBUTED AND EMBEDDED SYSTEMS workshop*, 2001.

[7] F. Zhao and L. Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Elsevier and Morgan Kaufmann Publishers, 2004.

[8] C. Savarese, J.M. Rabaey, and J. Beutel. Locationing in distributed ad hoc wireless sensor networks. In *Proc. 2001 Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP 2001)*, volume 4, pages 2037–2040, May 2001.

[9] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low cost outdoor localization for very small devices. Technical Report 00-729, Computer science department, University of Southern California, Los Angeles, CA, 2000.

[10] J. Beutel. *Master Thesis, Geolocation in a PicoRadio Environment*. UC Berkeley, 1999.

[11] A. Savvides, C. C. Han, and M. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *ACM SIGMOBILE*, 2001.

[12] A. Howard, M. Mataric, and G. Sukhatme. Mobile sensor network deployment using potential fields: A distributed solution to the area coverage problem. In *DARS 02*, 2002.

[13] Fortune. Voronoi diagrams and delaunay triangulations. In *Computing in Euclidean Geometry, Edited by Ding-Zhu Du and Frank Hwang, World Scientific, Lecture Notes Series on Computing – Vol. 1*. 1992.