MASmote – A Mobility Node for MAS-net (Mobile Actuator Sensor Networks)

Zhongmin Wang, Zhen Song, Peng-Yu Chen, Anisha Arora, Dan Stormont and YangQuan Chen
Center for Self-Organizing and Intelligent Systems (CSOIS)
Dept. of Electrical and Computer Engineering
Utah State University, Logan, Utah 84322-4160, USA

***Abstract*— This paper presents a new mobile robot platform (MASmote) acting as a sensing or actuation node with mobility in mobile actuator sensor networks (MAS-net). The hardware components and software architecture of MASmote are introduced in detail. The dynamic model of the MASmote is described and a low-level controller is designed to make the motion of these MAS-net nodes reliable and flexible. MASmote is built from commercial off-the-shelf components. Its small size, low cost, solid structure, and software modularity make it suitable for large-scale distributed mobile actuator and sensor network. The application of MASmote and MASmote system is exemplified in a scenario of diffusion process boundary tracing.**

***Index Terms*— mobile actuator sensor network (MAS-net), mobile robot, mobility platform, distributed system, low-level control, software modularity.**

I. Introduction

The developments in large-scale integrated circuit, low power consuming sensors and efficient wireless communication have advanced the application of wireless sensor networks. To achieve higher time and spatial resolution, sensor network built upon a mobile platform is desirable. By introducing actuators into the sensor network, the networked sensing capability can be extended with enhanced manipulation. But at the same time, the control strategy and collaborative signal processing algorithm become more challenging. Senor network suffers from the constraints of limited power supply, unreliable communication, and limited computation ability of nodes. To make the mobile actuator sensor network work properly, a reliable, yet low-cost mobile platform with enough functionality is essential.

Serval research groups have developed different kinds of mobile platforms for sensor networks [1], [2], [3], [4]. These robotic platforms displayed their functionality and flexility in experimental environments to some extent. Among them, MICAbot is a feasible robotic platform for sensor networks. Using Hall-effect sensor for its encoder, the MICAbot is sensitive to noise[4].

In this paper, we present a new mobile platform that is suitable for mobile actuator sensor network. It is compact, low cost, structurally robust and flexible in configuration. It has little requirements on hardware components, yet powerful enough to fulfill advanced tasks. Software modularity makes new applications easily developed. Because of its flexibility in software and its compact hardware structure, MASmote requires little maintenance in most application scenarios, which is an attractive feature for large-scale sensor network, where the maintenance can be a formidable job, if not impossible.

MASmotes are built mainly by commercial off-the-shelf (COTS) components. It comprises only one control board, one interface board, and a base platform which loads two 6 volts modified servo motors, as shown in Fig. 1. The control and communication board used in MASmote is MICA2 [5] board from CrossBow [6]. We use TinyOS [7], a popular event-driven operating system for embedded MEMS system, as the node operation system. All software components are developed using the NesC programming language. Unlike CotsBots, which implements an Ackerman steered wheel robot, MASmote is a two-wheel differentially steered mobile robot. It is structurally much simpler, yet still manipulation flexible. MASmote has digital IR reflector as its wheel encoders which can be used for odometry measurement and localization. With different sensors and/or actuators, the MASmote system can achieve different tasks. Currently, we are working on a diffusion-boundary-detection application, which is described in [8], [9]. The basic idea is to detect the boundary of a diffusion chemical, which is fog for our theory-validation testbed, with a mobile sensor network formed by MASmote.

Fig. 1. The prototype of MASmote platform – the mobility node of MAS-net (mobile actuator and sensor networks).

This work is supported in part by Space Dynamic Laboratory (SDL) "Skunk Works Fund", Utah State University. Corresponding author: Dr YangQuan Chen. E-mail: yqchen@ece.usu.edu; Tel. 01-435-7970148; Fax: 01-435-7973054. URL: http://www.csois.usu.edu/people/yqchen.

The remaining part of this paper is organized as follows. In Sec. II, the hardware design and software components are presented. MASmote characteristic and its low-level control is described in Sec. III. Sec. IV is devoted to the proposed MASmote system architechture with details on how the high level control strategy is implemented. To show a representative application for our MASmote, the monitor and control of diffusion process are discussed together in Sec. V. Finally, Sec. VI concludes this paper with some remarks on further investigations.

II. MASMOTE DESIGN DETAILS

A. Design Consideration

The mobile platform used in large-scale wireless actuator sensor network is required to be of small size and low cost, with easy software development and requires little maintenance needed during experiments. To deal with different application scenarios, even hostile environments, the platform should be structurally stable and not sensitive to disturbance.

The size of MASmote is suitable for experimental investigation of mobile actuator sensor network in a laboratory environment. The dimension of MASmote is only 9.5cm $\times$ 9.5 cm $\times$ 6.5 cm.

Low cost is another advantage of MASmote. Most hardware components in our mobile platform are commercially available. Little custom work is needed. Our central control board with wireless communication ability is from Crossbow.The motor is from [10]. The base chassis is from [11]. The total cost for each MASmote is about $\$250$. With the "programming on the fly" feature provided by TinyOS, we can update the program on each MASmote via wireless connections without physically linking the MICA2 board to the base station. Thus, the work load of maintained is reduced.

At the current stage, our first prototype of MASmote is task orientated. Only basic functionality is implemented so that MASmote is suitable for monitor and controlling of a diffusion process. These basic functionality includes onboard processing, wireless communication, motor control, and fog concentration level sensing. But it has powerful processing capability and sufficient storage memory to meet different application requirements. Only minor modification is required to extend its capability to a wide range of research scenarios. We have reserved connector for additional sensor board to be easily added.

B. MASmote Hardware Design

MASmote is an inexpensive mobile platform built entirely from commercial off-the-shelf components. We use MICA2 board as the control board for MASmote. Unlike other mobile platform design, it is the only controller in MASmote. MICA2 board is developed by CrossBow. It has ATmega 128L as its central controller. ATmega 128L [12] is a high performance, low-power consumption, 8 bit controller based on the AVR enhanced RISC architecture. ATmega 128L onboard works at 8 MHz. It achieves throughput approaching 8MIPS by executing multiple instructions in a single clock time. Powered by 3V, it is ideal for power constrained computation and control. ATmega 128L has 128 KB in-system programmable flash memory, 4KB EEPROM and 4KB SRAM. Node applications can be easily included in the onboard chip. It has reconfigurable PWM output which is used to produce PWM signals to motors in MASmote. It also has 8 10bit ADC channels which can be used for multiple sensors on the MASmote. ATmega 128L has serval configurable timers/counters and multiple data interfaces including I2C, SPI and UART. Although only a few sensors are equipped with MASmote, more sensors can be added to MASmote with minor modification. In this case, an additional sensor board may be needed.

Integrated in the MICA2 board is the CC1000 [13], a single chip, low power RF transceiver. It works at 916MHz in MICA2. The FSK data rate can be as high as 76.8K Baud and the wireless range is from 10 ft to 100 ft. This reprogrammable transceiver has the appealing feature of low power consumption desirable for power constrained wireless applications. ATmega 128L and CC1000 form the kernel of MICA2 board.

MASmote is structurally compact. Less hardware makes MASmote reliable and stable. The MICA2 board, interface board, motors, and battery holder are assembled in one chassis. We adopt the Mark III chassis kit [11] for MASmote body.

Two modified servomotors are used to drive the MASmote. A servo controller is encapsulated in motor components. No additional H-bridge is needed. Each motor has a 5 pin ribbon cable connected to the interface board with one pin for the PWM signal, two for the direction control, and two for the power supply. The servomotor can provide 56 oz-in of torque at 6 volts, which makes it suitable for rough surface. The wheels are 6.6 cm in diameter. Controlled by full duty cycle PWM signal, the servomotors can drive MASmote at the maximum velocity is about 25 cm/sec.

Onboard odometry measurement is essential for an autonomous robot. But it requires accurate measurement sensors. Because of its high cost, commercial encoder is not suitable for mobile platform in large-scale sensor networks. Thus, we decided to make a simple, low-cost encoder wheel for MASmote. The encoder pattern is drafted using Matlab. It is printed and glued onto inner side of the wheels. The encoder has 60 segments. This number is determined by experiments, which gives MASmote a resolution of 6 deg/strip, or about 0.3cm in length. The encoder pattern we used is shown in Fig. 2.

Two photo-reflectors [14] are mounted on the platform body as encoders. They can detect white band and black band on wheel encoders with digital output. The grey level detection characteristic is shown in Fig. 3.

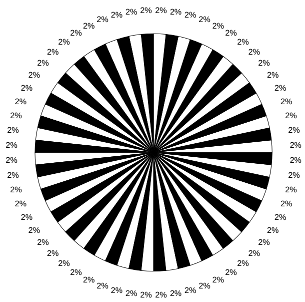

Fig. 2. Encoder pattern used for MASmote.

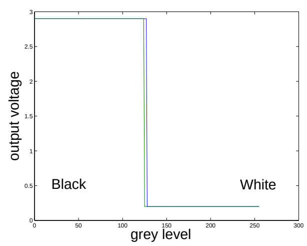

Fig. 3. Grey level detection input-output curve (3 Volts power, 0.2 cm distance).

It can be seen that the photo-reflector works digitally. It is not sensitive to noise. Its performance is satisfactory even under uneven illumination.

There is one IR range detector mounted in the front of MASmote, which can be used to provide the ability of obstacle avoidance. It has an effective detector range of 10 to 80 cm. Two photoresistors on the bottom are used to measure the fog concentration in our experimental testbed (See Sec. IV).

C. MASmote Software Design

Software on MASmote is based on TinyOS operating system. The event-driven execution model is related with interruption handling occurred in hardware. Event-driven execution enables fine-grain power management and flexible scheduling which are necessary for unpredictable wireless communication and physical environment. The component-based architecture of TinyOS enables software reusable. The user can extend components by implementing new components for a specific application. Since the hardware configuration on MICA2 is fixed, many components, such as communication and ADC sampling are already provided. But they can be further redefined for specific purposes.

According to [15], "*There are two threads of execution in TinyOS: tasks and hardware event handlers. Tasks are functions whose execution is deferred. Once scheduled, they run to completion and do not preempt one another. Hardware event handlers are executed in response to a hardware interrupt and may preempt the execution of a task or other hardware event handler*." TinyOS system and applications were written in NesC, a C-like language for programming structured component-based applications and supports the concurrency model (task and event) of TinyOS [16].

We developed our application on MASmote by using modular software components. At the highest level, they are combined together to build a single application. The layered component structure is shown in Fig. 4. Only the "main application" component needs to be modified, if necessary, for different experiment purposes. The XNP is a component from TinyOS component library. It is responsible for in-network programming of MICA2 using radio communication [17]. When programs are loaded into MASmote, modification or upgrading of program can be made on the fly. It is a convenient way to change or repair the MASmote functionality. Since the TinyOS is not deliberately designed for mobile platform, its original component library does not contain components that can produce PWM signals to drive DC motors. We designed our own PWM component by using NesC. This component works in the fast PWM Mode. Its PWM period and duty cycle are adjustable.

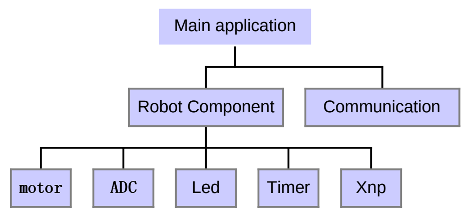

Fig. 4. Component architecture in MASmote.

The capability of wireless communication of MICA2 has been thoroughly exploited in our MASmote. We use different message types for different tasks. Sensor information and inter state of MASmote are interchangeable among different MASmotes or between each MASmote and the base station. The base station can even modify the low-level controller parameters in MASmote by sending a message via radio. It makes the whole system very flexible and easy to maintain.

III. MASmote Characteristic and Low Level Control

A. System Modelling and Control

Open-loop control experiments have been made to investigate the MASmote's performance under typical lab setting. Speed is set by specifying PWM duty cycles from 0 to 100 percent. The experiment shows that the relationship between the PWM input and the speed is not linear, but close. The PWM period is another factor that affects the speed response performance of MASmote, especially in lower speed, as can be seen from Fig. 5.

The PWM periods of 2 ms, 10 ms and 20 ms were used in our experiment. The experiment shows that the PWM signal with a 20ms period has a better performance at both the low and high duty cycles. The output wheel velocity is also affected by the battery voltage. But this deviation can be

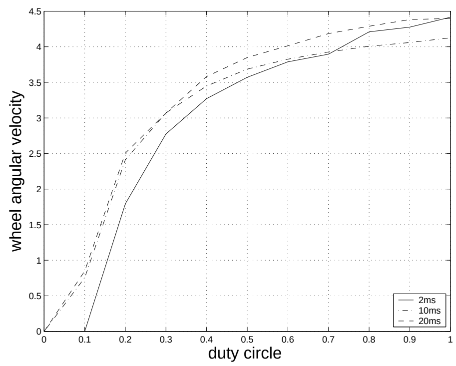

Fig. 5. Speed response performance under different PWM periods.

compensated by measuring battery voltage via ADC channel-0.

The dynamic model of the motor is approximately obtained as a first order system

$$G(s) = \frac{k}{\tau s + 1}. \tag{1}$$

The parameters of the model, k and τ, can be obtained by a velocity step response. The values obtained from experiment are $k = 1.1$, $\tau = 0.05$, respectively, as shown in Fig. 6.

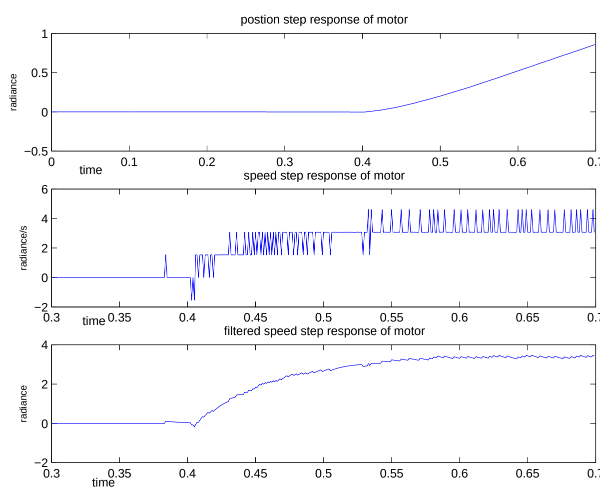

Fig. 6. Velocity step response of a MASmote motor.

Since the velocity step response is not directly available, we measured its position response and then differentiated the data. The differentiated data was filtered by a low-pass filter,

$$G(s) = \frac{1}{1 - 0.93z^{-1}}. \tag{2}$$

The kinematics of a two-wheel robot like MASmote is shown in Fig. 7(cited from [18]).

Assume that the sampling interval in MASmote is $\triangle T$, we can derive the kinematic discrete time model as follows:

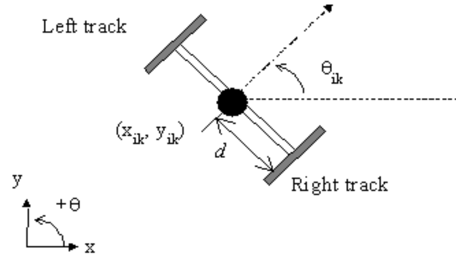

Fig. 7. Cart model.

$$x(k+1) = x(k) + \frac{v_l + v_r}{2}\cos(\theta(k))\triangle T \tag{3}$$

$$y(k+1) = y(k) + \frac{v_l + v_r}{2}\sin(\theta(k))\triangle T \tag{4}$$

$$\theta(k+1) = \theta(k) + (v_r - v_l)\triangle T/d \tag{5}$$

where v_l and v_r are the line velocities of left wheel and right wheel, respectively.

Based on the dynamic model of MASmote, we can develop the low-level controller of the MASmote. First, we consider the position feedback of MASmote. Since the dynamic model of the motor is not very accurate, and also because there are nonlinear blocks in the control loop, we adopt the widely used PID control algorithm. Here we only consider the P controller. The parameter of the P controller is designed to put the roots of the system characteristic equation to the left half s-plane. To evaluate the impact of nonlinear blocks, simulation model was designed to validate of our position feedback controller. Consequently, the P value is finely tuned by the simulation result. The simulation model is shown below:

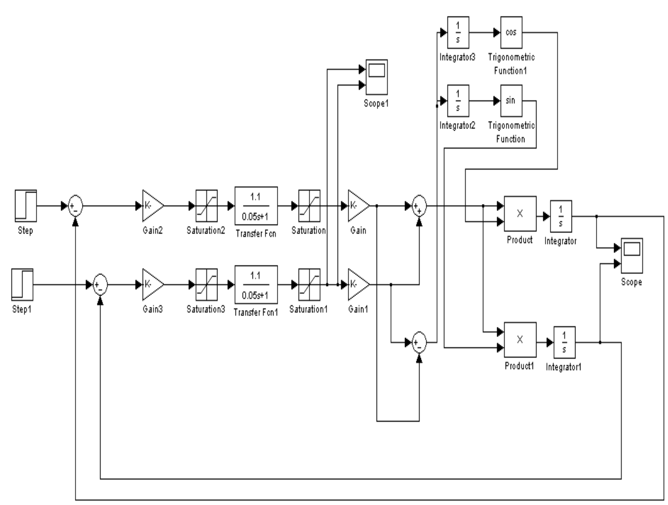

Fig. 8. The Simulink model of position feedback control.

As shown in Fig. 8, there are two nonlinearities in the control system. One is from the saturation block. The reason is, for battery-powered MASmote, the maximum voltage supply is 6 volts. On the other hand, the maximum angular velocity of the motor is limited to 5 radians/second, as observed from our experiments. The other source of nonlinearity is the trigonometric manipulation in the closed loop. The optimal theoretical value of P is hard to find and can only obtained by trial-error. By designing the software wisely, we can change the value of P controller on the fly from the base station, as mentioned above. Different mote may have different P controllers.

IV. MAS-NET SYSTEM ARCHITECTURE AND HIGH LEVEL CONTROL

A. MAS-net System

Our proposed MAS-net system is comprised of serval components, as listed below:

- Base station: Provides system coordinate.
- Vision subsystem: Provides pseudo-GPS information (position and orientation).
- MASmotes: Ten MASmotes in our current system.

An illustrative map is sketched in Fig. 9.

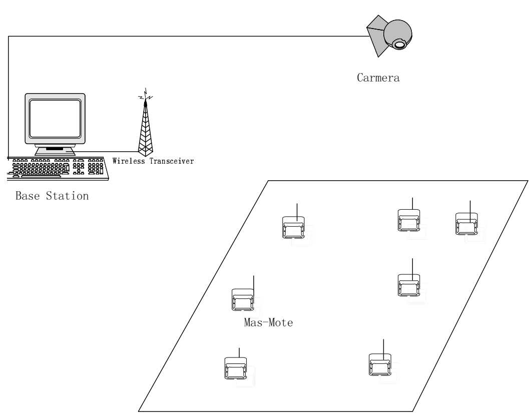

Fig. 9. Components of MAS-net system.

Based on the mission to be implemented, there are two control frameworks to be used. One is the decentralized control, and the other is the centralized control.

For the decentralized control, every MASmote makes a decision on all the information it can obtain. But, it dose not receive command from the base station. Base station only provides pseudo-GPS information to MASmotes. Co-operative mobile multi-agent system can be built in this way. For the centralized control, MASmote is responsible for its low-level control and information gathering. The base station makes control decisions based on the information from the whole system and send different control commands to individual MASmote. MASmotes can communicate with each other directly. It is desirable to have in decentralized control system an option of centralized control.

B. Vision System

Due to the dead-reckon feature of the encoders, the position-estimation error is approaching to infinity along the time. Thus, we need to calibarte the odometry system regularly. In our testbed, a CCD camera is used to provide a pseudo-GPS service to the MASmote. Every MASmote in scope has a label attached on top of it. An algorithm is under development to the extract MASmote position and orientation from the images. The base station can send MASmote its coordinate information every 500 ms via the wireless communication for calibration. When the accurate position information is available, it will be used in the feedback control loop and to estimate the position of further movement until new data from the base station is arrived.

V. MASMOTE FOR DIFFUSION PROCESS CHARACTERIZATION

In this section, we describe how the MASmote is expected to be used in diffusion process tracing and zone control. This research is motivated by three proposed application scenarios: [9]

- The safe ground boundary determination of the radiation field from multiple radiation sources.
- The nontoxic reservoir water surface boundary determination and zone control due to a toxic diffusion source.
- The safe nontoxic 3D boundary determination and zone control of biological or chemical contamination in the air.

In the current stage, we focus on the case of a 2D diffusion process and use a team of ten MASmotes to track the diffusion process of fog. The emission of the fog through a small outlet is adjusted in such a way that the dilation speed is relatively slow. The MASmote is expected to keep on moving and distributed evenly along the boundary of diffusive fog. This diffusion boundary determination is formulated as model-based distributed control task. The mathematical model of the diffusion processes can be described by the following parabolic type partial differential equations [9]:

$$\frac{\partial V(q,t)}{\partial t} + \nabla \cdot (FV(q,t)) = \nabla \cdot (D(q,t)\nabla V(q,t)) + g(q,t) \tag{6}$$

$$V(q_0, t_0) = v_0 \tag{7}$$

where $\nabla^2 = \Delta$ is the Laplace operator; $q = (x, y, z)^T \in \mathbb{R}^3$ is the spatial variable; $V(q,t)$ is the spatial-temporal distribution function; $FV(q,t)$ denotes the external inputs on the plant dynamics (e.g., wind, rain, dust, humidity, etc.); $D(q,t)$ is the carrier diffusion function for the specific problem; $g(q,t)$ reflects the effects of constraints (e.g., gravity). V_0, q_0, t_0 denote the initial conditions.

Consider the problem of boundary tracing in MASmote system. The goal is $\hat{V}(q_i, t^*) \rightarrow V(q_i, t^*)$ for all i. The objective function can be formulated as

$$J = \sum_i (V(q_i, t) - \hat{V}(q_i, t))^2, \tag{8}$$

where i is the index of MASmotes. $\hat{V}(q_i, t))$ is the estimated fog density based on sensor measurement.

The general control framework of MASmote based MAS-net system can be described in Fig. 10 [9], [8].

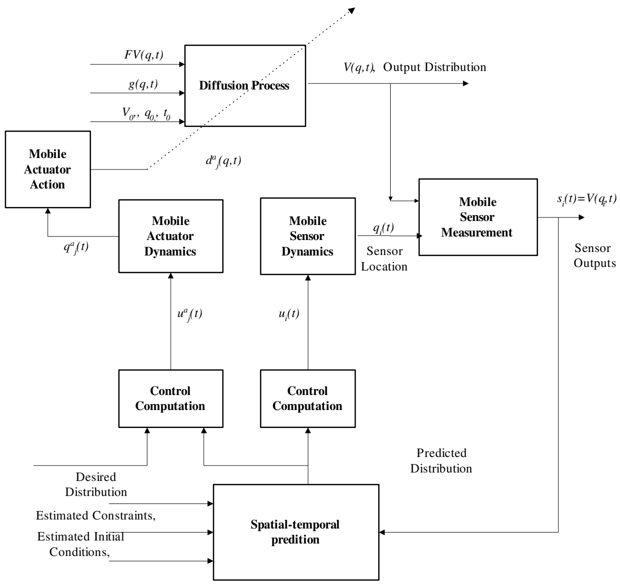

Fig. 10. Block diagram of MAS-net system [9], [8]

As shown, the sensor of the i^{th} of MASmote, located in space at $q_i(t)$ takes samples, denoted by $s_i(t)$, which by definition is $s_i(t) = V(q_i, t)$. One way to obtain the measurement is to scan the focused area line by line. The measurements from all the sensors are then used by the predictor, along with estimates of the media flow, initial conditions, and constraints, to produce a predicted distribution, $\hat{V}(q, t)$. This is then used by a controller to decide on the control input $u_i(t)$ to drive the MASmotes. Based on the control strategy adopted, motion command is sent to the MASmotes and MASmotes will drive to the desired place to sample the fog concentration there. They will then keep on the boundary of the fog. Discussions about the control-theoretic aspects of this problem can be found in [9].

VI. Conclusions

A novel mobile platform for mobile actuator and sensor network (MAS-net) is presented in this paper. The MASmote is a low-cost, flexible, modular robots for experiment investigation in sensor network. Equipped only with one MICA2 board, one interface board and two servo motors in a small size chassis, it is hardware compact, yet powerful enough for challenging high-level tasks. All the applications are developed under TinyOS. Component-based software makes new algorithms easy to implement. When program is downloaded into MASmote, the only manual work is to change the battery if necessary. Code upgrading can be done by programming via radio communication. MAS-net can be built easily to accomplish various high-level tasks with our MASmotes.

VII. Acknowledgements

We would like to thank Dr. Kevin L. Moore, the Past Director of CSOIS, Utah State University, for his initiation of the MAS-net project. We would also like to thank Bruce Christensen and Lili Ma for their help with the project.

VIII. References

[1] S. Bergbreiter and K.S.J. Pister, “Cotsbots: An off-the-shelf platform for distributed robotics,” in *in Proc. of the IEEE Int. Conf. on Robotics and Automation*, Las Vegas, NE, October 2003.
[2] G. Sibley, M. Rahimi, and G. Sukhatme, “Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Washington D.C, USA, May 2002, vol. 2, pp. 1143–1148.
[3] Robert Grabowski, Luis E. Navarro-Serment, Christiaan J. J. Paredis, and Pradeep K. Khosla, “Heterogeneous teams of modular robots for mapping and exploration,” *Autonomous Robots*, vol. 8, no. 3, pp. 293–308, 2000.
[4] Mb. Mcmickell, Bill Goodwine, and L. A. Montestruque, “MICAbot: A robotic platform for large-scale distributed robotics,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2003.
[5] Crossbow, “MICA2,” `http://www.xbow.com/Products/productsdetails.aspx?sid=69`.
[6] Crossbow, “Crossbow,” `http://www.xbow.com/Products/productsdetails.aspx?sid=3`.
[7] “TinyOS,” `http://webs.cs.berkeley.edu/tos/`.
[8] Kevin L. Moore, Yang Quan Chen, and Zhen Song, “Diffusion based path planning in mobile actuator-sensor networks (MAS-net) - some preliminary results,” in *Proc. of SPIE Conf. on Intelligent Computing: Theory and Applications II, part of SPIE’s Defense and Security*, Orlando, FL, USA, Apr. 2004.
[9] YangQuan Chen, Kevin L. Moore, and Zhen Song, “Diffusion boundary determination and zone control via mobile actuator-sensor networks (MAS-net): Challenges and opportunities,” in *Intelligent Computing: Theory and Applications II, part of SPIE’s Defense and Security*, Orlando, FL, Apr. 2004.
[10] Solarbotics, “Solarbotics,” `http://www.solarbotics.com/`.
[11] Junun, “Mark III Robot Store,” `http://www.junun.org/MarkIII/Info.jsp?item=2`.
[12] ATMEL, “Atmega 128(L) Preliminary,” `http://www.atmel.com`.
[13] Chipcon, *SmartRF. CC1000 preliminary datasheet*, rev. 2.1 edition, April 2002.
[14] Hamamatsu, “Photoreflector P5587, P5588,” `http://www.hamamatsu.com`.
[15] TinyOS, “TinyOS tutorial,” `http://webs.cs.berkeley.edu/tos/tinyos-1.x/doc/tutorial/`.
[16] David Gay, Philip Levis, David Culler, and Eric Brewer, *NesC 1.1 Language Reference Manual*, May 2003.
[17] Inc. Crossbow Technology, *Mote In-Network Programming User Reference*, March 2003.
[18] “The Millibots project,” `http://www-2.cs.cmu.edu/~cyberscout/localization.html#system`.