ragobots: Real Action Gaming Robots

1. Research Team

Faculty:	Prof. Mani Srivastava, *Electrical Engineering, UCLA* Prof. Deborah Estrin, *Computer Science, UCLA* Prof. Gaurav S. Sukhatme, *Computer Science, USC* Prof William J Kaiser, *Electrical Engineering, UCLA*
Staff:	None
Graduate Students:	Aman Kansal, *Electrical Engineering, UCLA* Jonathan Friedman, *Electrical Engineering, UCLA* Parixit Aghera, *Computer Science, UCLA* Balaji Vasu, *Computer Science, UCLA* Advait Dixit, *Computer Science, UCLA* David Lee, *Electrical Engineering, UCLA*
Undergraduate Students:	None
Industrial Partner(s):	None

2. Overview

An emerging area of research in embedded systems is the use of actuation and controlled mobility. Autonomous devices capable of performing varying degrees of self-initiated motion can be introduced into a sensor network to provide several performance advantages such as enhanced coverage, resource redistribution, zooming-in capability for phenomenon of particular interest, repairing communication connectivity, fault detection, calibration and localization. This opens up several research challenges in developing methods for determining the use of motion and providing navigational and other support for autonomous controlled mobility. Real Action Gaming robots (ragobots) is a laboratory scale test-bed for exploring these research issues in a fun setting for students.

Ragobots also provides an exciting application for embedded systems and is an important educational aid for introducing students at all levels to sensor networks and robotics.

3. Approach

We are developing a network of mobile robots with basic navigational support to carry out various robotic activities in a sensor network. The test-bed is designed as gaming platform to expedite development and encourage creative ideas. This is based on established psychological results which show that while at play, human beings

- Seamlessly move in and out of the context of play

- Employ a separate mental cognition

This relaxed sense of cognition and ambiguous context opens the mind to fanciful interpretations and creative ideas. The sense of play also makes people less averse to accepting new beliefs and learning new skills compared to performing the same activities for work. For example, people study travel literature, complex maps and browse through incomplete information to plan vacations with ease while they may have hard time browsing a much more structured and complete datasheet to design a work related system. We hope to recreate this feeling of “play-time” in sensor networking and robotics research by creating a research platform built around the notion of gaming.

The test-bed itself is designed with all the fundamental capabilities required for the development of autonomous actuation. The robots have mechanisms for localization and navigation. They can detect the presence of interesting objects in the environment and take responsive actions. The prototype robots built for the test-bed can use range-sensors to learn the presence of significant terrain features which can be used to develop sophisticated mobility and placement planning algorithms.

The test-bed is so designed as to allow the experimenter to isolate the functionality of one algorithm while other dependent components are run through a central server to develop and test systems one module at a time.

The basic games are designed to have similar goals as the performance metrics in networked sensing, such as assuring maximum coverage and tracking phenomenon of interest. A high level view of the platform is given in Figure 1.

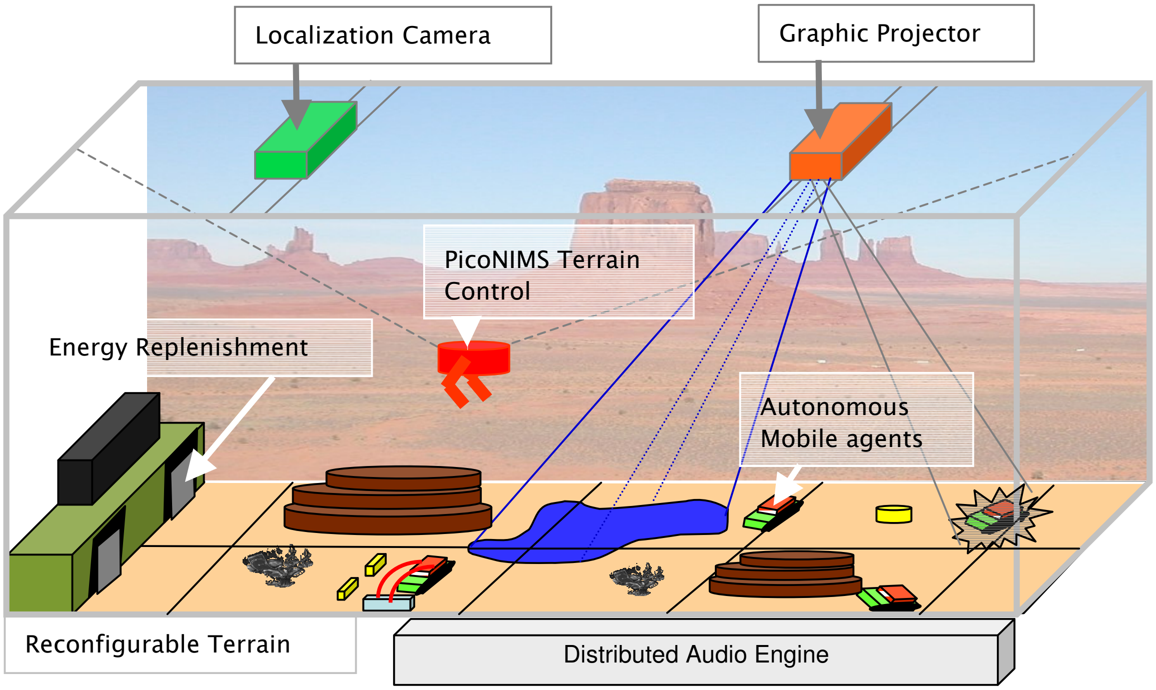

Figure 1. Ragobot platform overview.

4. Experiments and Systems

The test-bed prototype consists of several entities, consisting of the robots, the terrain on which the robots move, a central server which controls the robots and provides localization and other experimentation services. These are described in more detail below.

4.1 Hardware

This consists of several entities, labeled workerbots, marinebots, structures, resources, localization infrastructure, PicoNIMS enabled terrain reconfiguration infrastructure and static terrain components. These components are discussed below.

1. **Workerbots:** These are robots, consisting of following components (Figure 2)
   a. a processor and radio platform: MICA2
   b. a twin-DC-motor enabled traction mechanism to move on a 2D surface with small undulations
   c. two servos for pan and tilt control for a pointing device
   d. tilt sensor
   e. an IR range sensor (emitter and detector) mounted on the pan and tilt
   f. an object gripping mechanism

g. Battery Monitoring Circuit
h. Battery Charging circuit
i. Contact sensor mounted on gripper to detect resource

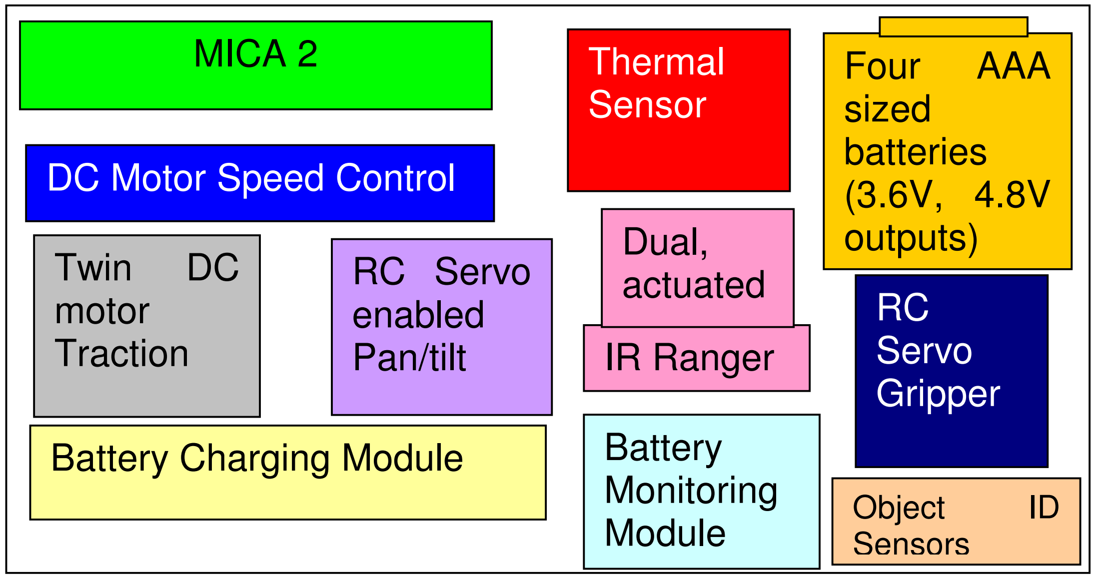

Figure 2. Workerbot system block diagram.

2. **Marinebots:** These are a workerbot, (optionally without the servo controlled gripper) but with a simulated weapon system added:
    a. IR emitter with embedded digital signature to reveal identity of firing robot and the class of weapon used (the class of weapon depends on resources spent and influences the extent of damage in the victim entity)
    b. IR detector to detect when "hit" by another robot's IR emissions. This detector may be directional.

3. **Structures:** These components represent stationary structures available to the player in the game. These are locations for collecting resources. The structures can be damaged by firing from enemy robots and can be repaired by collecting sufficient resources. They consist of:
    a. Mica2 processor and radio platform
    b. IR emitter, strategically pointed at resource collection point for robot to acknowledge deposition of a collected resource.
    c. IR weapon sensor to sense damage when hit
    d. RC Servo enabled collapsible and repairable walls/roof.

4. **Resources:** These are detectable objects which are deigned to represent various resources to be collected by the player during the game.
5. **Terrain:** This consists of a custom-tiled floor with obstacles and undulations, some of which block the robots and some allow the robot to pass.

6. **Localization infrastructure:** This consists of an overhead web-cam connected to a PC. Each robot has a special pattern pasted to its top which helps the image processing algorithm to recognize the various robots on the terrain.
7. **PicoNIMS enabled terrain reconfiguration infrastructure:** This consists of two standard PicoNIMS nodes with added gripper functions. These can add, remove and relocate components in the player terrain.
8. **Distributed Speaker System:** A set of 4 to 6 speakers, placed across the terrain (embedded underneath the terrain floor) for generating sound effects at relevant places.
9. **Overhead Graphics Projector:** This is an LCD projector mounted to have an arieal perspective of the terrain and is used to generate visuals on the terrain.

The range sensors and weapon detection sensors together enable the robots to detect obstacles and other IR emissions. This could be used for

1. sensing uncertainty control
2. obstacle sensitive navigation
3. obstacle sensitive deployment
4. LoS sensing range based coverage topology management
5. target detection (with targets modeled by IR emitters, range can be controlled by selecting detection thresholds)

A picture of the prototype robot is shown in Figure 3.

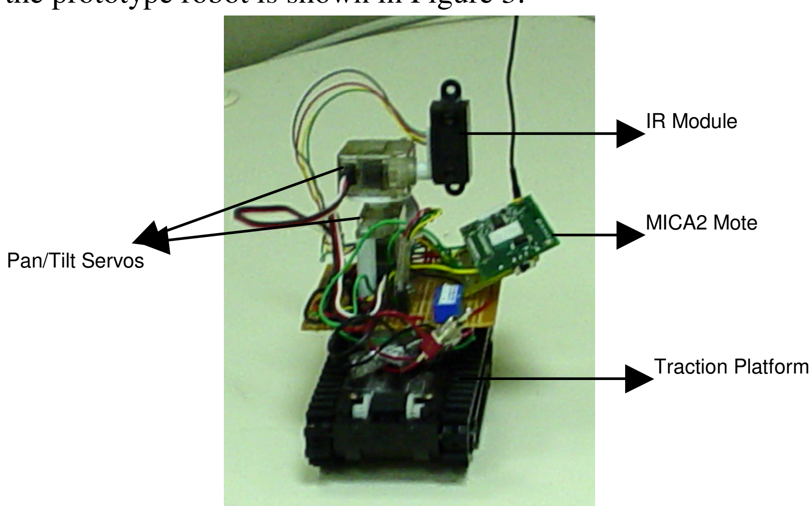

Figure 3. Prototype robot.

The first component being developed is the workerbot. The initial hardware design of this robot, shown in Figure 3, is discussed next. The Ragobot robot consists of four hardware layers. The upper-most, observation, layer hosts a dual-axis pan-tilt actuator platform for positioning a spotlight, infrared ranging sensor, and any sensor needed in the future. The next layer down, Ragobot, is the primary controller board. The Ragobot layer circuit board hosts the physical interface to the Berkeley Mote (which provides processing and communication facilities), 27 LED's in various configurations and colors (for entertainment and status display purposes), an advanced power supply, a battery charger, programming and expansion headers, and a dual DC motor controller module (an off-the-shelf solution provided by Polulu, Inc.).

Below the Ragobot layer is the Redloc layer, which is presently undeveloped. Redloc will host an sparse 2D array of infrared receivers in addition to a supplemental sensor suite for the robot (odometers, compass, and attitude sensor). The redloc layer's IR array will use a beam-forming approach to provide simulated weapons fire between robots as well as provide point-to-point communications links for location specific tasks such as signaling that the robot has arrived at the hangar.

The lowest layer of the Ragobot is a commercial traction platform which makes use of two DC motors to drive tank treads through a three gear, fixed-ratio drivetrain. The traction platform's motors are connected through the Ragobot layer to a Polulu motor controller, which provides the Berkeley Mote the ability to command the platform to move at over 100 different speeds each in both the forward and reverse directions.

Since the Ragobot contains RF electronics, analog sensing circuits, and digital logic, significant design effort was required to ensure that noise injection between sources is minimized. If the traction platform is powered directly from the on-board battery, severe fluctuations in the power rail supplying the radio, sensing, and digital electronics appear (Figure 4).

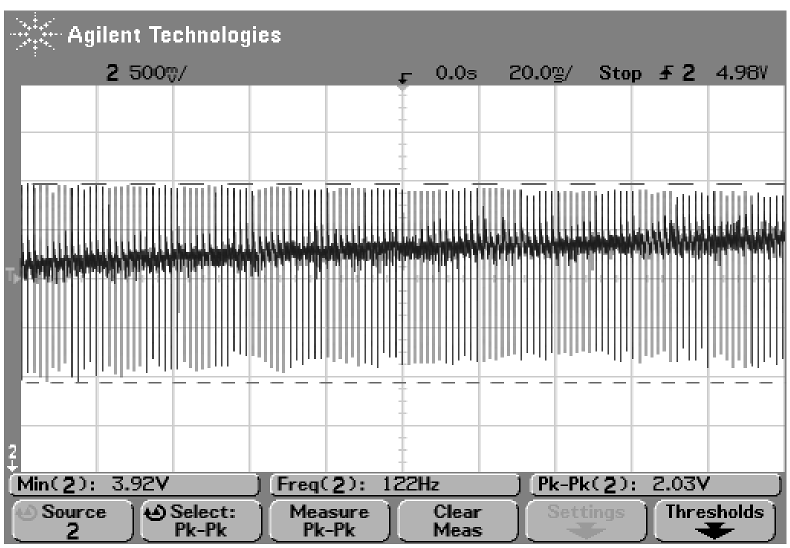

Figure 4. Power supply characteristics without filtering out noise.

To solve this problem a sophisticated power supply and filtering network were employed. Analog simulation of the Ragobot running at full speed while simultaneously sensing and transmitting on the radio (i.e. worst case) reveals that our filter performance is sufficient to provide an RF quiet operating environment (Figure 5).

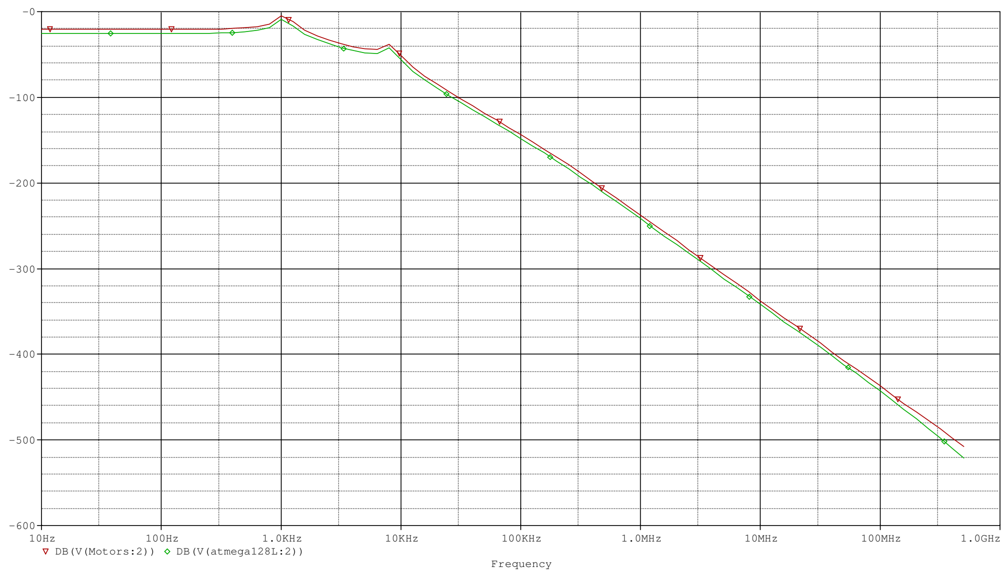

Figure 5. Frequency Response of power supply filters.

4.2 Software

Software for Ragobot is designed using component based framework provided by TinyOS and nesC. NesC is programming language designed for event-driven embedded systems. Following are the design goals for our software design.

Design Goals

Modularity: Software component should be designed in modular way to enable software reuse and enhancement easy.

Efficiency: Since we are implementing software in very constrained processing environment, efficiency is an important issue. The software design should be efficient in terms of RAM usage, code space usage, processing time.

Hardware Independence: It should be easy to change underlying hardware of Ragobot. To achieve this goal, interaction with hardware should be abstracted in a small number of driver components, which provides a hardware abstraction layer. All other components should use interface provided by driver component.

Real-time constraints: Real-time nature of the application puts real-time constraints on processing time. Software design should take these real-time constraints into consideration.

Figure 6 provides component interface diagram of Ragobot software. These component interfaces corresponds to nesC interface definitions. The components interfaces are divided in following two layers to achieve the modularity and hardware independence goals.

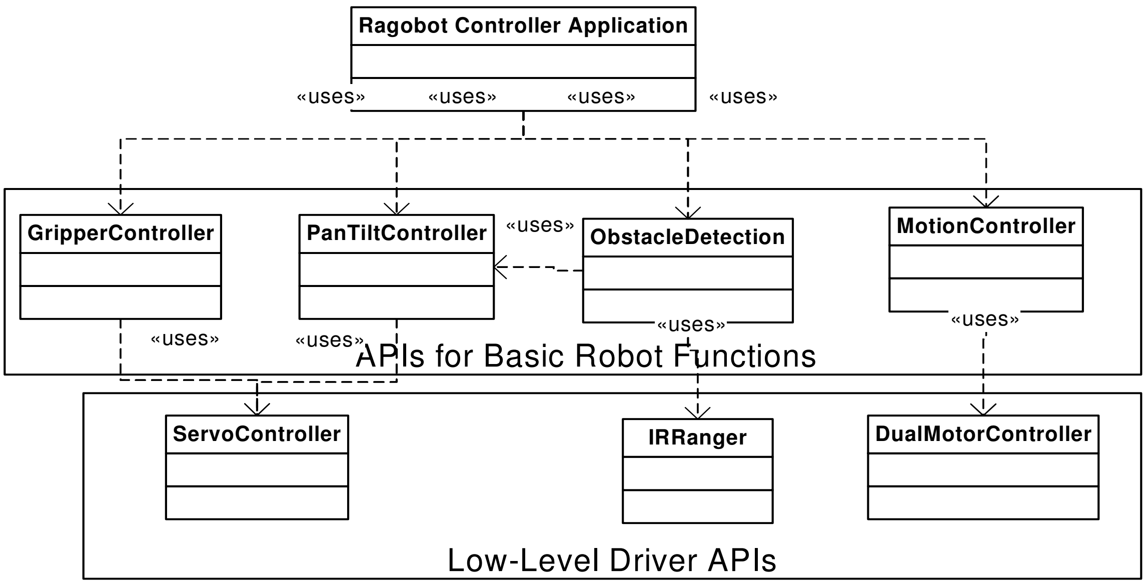

Figure 6. Robot software overview.

The lower level interfaces layer provides interfaces for different hardware components. Implementation of the driver for a particular hardware should provide appropriate interface specified at this layer. For example, change in a different hardware for IR ranging can be transparently achieved by modifying the IR ranger driver.

5. Problems Encountered

The initial problems included various mechanical construction problems, and testing software without a fully functional hardware platform. A first version of the hardware is now working thus making software development easier.

The image capture library used for the overhead camera based localization system was not available for the particular hardware being used in our prototype and hence a modified version of it is being developed.

6. Major Accomplishments

Key accomplishments in the first phase of the project include:

1. Initial hardware was designed and a PCB has been fabricated for it.
2. The first version of robot software has been developed which provides basic drivers for all hardware components.
3. A prototype robot is functional and is being used for testing various components.

4. An image processing platform for overhead camera based localization has been decided and is under development.
5. Education objectives realized:
    a. A course project based on this project was offered in CS213: Distributed Embedded Systems to help graduate students learn about embedded actuated platforms.
    b. A course projects based on ragobots has been offered in EE 206: Mobile and Wireless Communications to expose students to mobile communication systems which exploit controllably actuated elements.

7. Publications and Presentations.

1. A Kansal, “Robotic Gaming,” CENS Retreat, January 2004.
2. A Kansal, J Friedman, P Aghera, A Dixit, and B Vasu, “ragobots: Real Action Gaming Robots,” NIMS Meeting, March 17, 2004.

8. Future Directions

The ongoing and future work on this research project includes:
1. The prototype hardware developed is currently being tested.
2. Higher level functionality for the robot software is being developed.
3. The overhead camera based localization system is being developed.
4. Algorithm development for games and various experiments are underway.