MILO: Personal Robot Platform

Behnam Salemi

Department of Computer Science
USC/Information Sciences Institute
4676 Admiralty Way, Marina del Rey, Ca.90292USA
salemi@isi.edu

Jorge Reis, Arash Saifhashemi, Fred Nikgohar

RoboDynamics
2215 S. Sepulveda Blvd
Los Angles, Ca 90064., USA
{jorge, arash, fred}@robodynamics.com

***Abstract* – Recent advances in mobile robotic technology has given rise to new types of robotic applications called Personal Robotics. The main feature of these applications is that they require low cost robots that involve assisting people in doing their everyday jobs such as providing personal assistance, commercial security, mobile tele-conferencing, tele-presence, home services, entertainment, and elderly care. In this paper we will introduce a low cost and general purpose mobile robotic platform called MILO. It offers unique hardware and software features which make MILO suitable for a wide spectrum of personal robotic applications. MILO possesses PC architecture and runs on Windows XP. We have experimentally evaluated MILO's performance. Preliminary results show that MILO can be effectively used to develop personal robotic applications.**

Index Terms* – *Mobile robots, Internet robots, Personal robots, Service Robots, Tele-presence.

I. INTRODUCTION

Since early 1960's, *industrial robots* have been used in factories to perform tasks such as welding, cutting, drilling, painting, and parts assembly, all of which are either tedious or dangerous for human or require a high degree of accuracy. Industrial robots work in highly structured environments specifically designed for them which limit their interaction with humans. In the late 60's *mobile robots*, such as Shakey [1], gave rise to the concept of robots that are capable of performing tasks in unstructured environments and interacting with people in unison. This idea initiated numerous fields of robotic research such as: human robot interaction [2], simultaneous localization and mapping [3], distributed robotics and multi-robots systems [4], modular and self-reconfigurable robots [5], swarm robotics [6], etc.

Although advances in robotic research have made robotic technology a viable solution for many challenging problems such as space explorations [7], military operations [8], robotic surgery [9], and more, these applications often target specific problems, are very costly, and are mainly research oriented.

However, recent advances in robotic technology and the widespread use of the internet have given rise to new types of applications for mobile robots called *personal robotic applications*. The personal robotic applications are mainly applications that involve assisting people in doing their everyday jobs such as providing personal assistance, commercial security, mobile tele-conferencing, tele-presence, home services, entertainment, elderly care, and acting as learning tools. Examples of the available personal robots are iRobot Corporation's *Roomba* vacuum cleaner [10], ActivMedia Pioneer and PeopleBot [11] service robots, Evolution Robotics ER1 [12], and entertainment robots such as the Sony *Aibo* [13], and the Wow Wee *RoboSapien* [14].

Requirements of these new applications have raised new challenges for robotic research community which includes developing robots that are cost effective, reliable, safe, efficient, flexible, expandable, secure, and easy to program, maintain and interact with. Successful solutions to these challenges require novel designs that take issues such as expandability, efficiency, extensibility, modularity, and reusability of the software and hardware components into account.

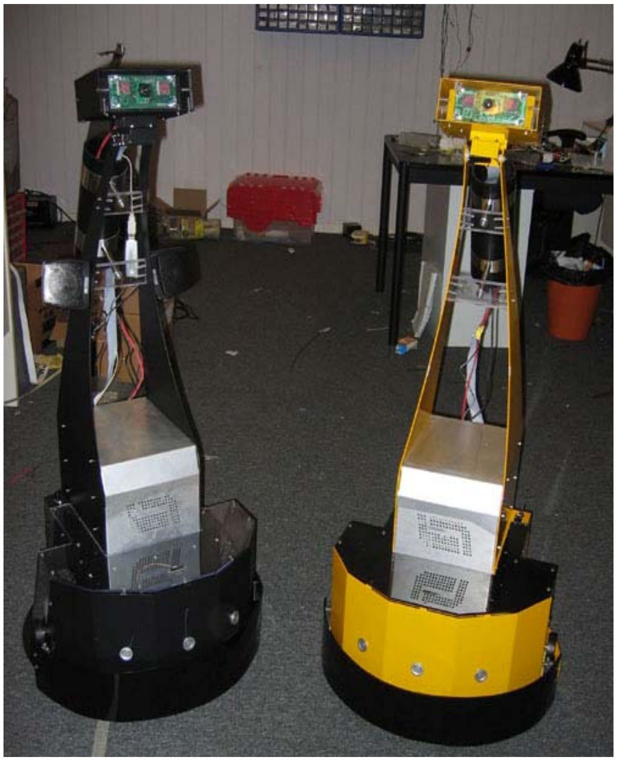


Fig. 1 MILO Personal Robot Platform

In this paper we introduce a new general purpose and low cost mobile platform called MILO that is designed for a wide spectrum of personal robotic applications such as commercial security, mobile tele-conferencing, tele-presence, personal assistance, elderly care and robotic research; see figure 1.

In order to reduce costs, MILO is built with off-the-shelf hardware and software components while proprietary components are minimal. A simple mechanical design and form factor are the unique features of MILO that make it an effective platform for the types of applications it is intended for. MILO also possesses PC architecture enabling it to utilize any and all PC accessories. Finally, MILO runs on Windows XP enabling it to fit into existing IT infrastructure, thus reducing the cost of development and support. Like a regular PC, users can remotely login to MILO to establish an audio-visual link and remotely control the robot via LAN or the Internet.

This paper is organized as follows: In section two we discuss the considerations in designing MILO. In Section three we will describe its hardware architecture. In section four we will describe the software architecture. Section five describes some performance evaluation experiments; and section six will conclude and describes the future research direction.

II. Design Considerations

Five criterions were considered in the construction of MILO. First, as MILO is intended to closely interact with people, safety is paramount. It was therefore determined that MILO should be cognizant of its environment through a series of sensors which prevent it from running into people and objects or falling off edges. The sensory information is also transmitted to the remote control interface to provide the user feedback about the current position and conditions of the robot.

Second, developing a low cost platform was another design consideration. This was achieved by using off-the-shelf hardware and software components where possible, and simplifying the mechanical design and fabrication materials.

Taking into consideration the intended applications for MILO, form factor was the third important issue of production. MILO is designed to stand by a table and participate in for example teleconferencing and/or to read documents remotely. For commercial security applications, it was important to position the camera at an appropriate height (approximately 105 cm) and as well as to give it the dexterity to navigate industrial ramps (15°). In applications that require direct interaction with humans such as personal assistance and elderly care, it is also important that the shape of the robot not be threatening.

Fourth, expandability, reliability, flexibility and ease of maintenance were of importance in MILO's design. A modular mechanical and electronic design allows for the extension of the robot for other applications such as remote sensing, manipulation ability and vision. In the current design, users have access to multiple general-purpose digital and analogue I/O ports through an open source firmware and a Windows based Application Program Interface (API).

Finally, a simple and effective graphic control interface was necessary for successful and safe use of MILO's resources such as audio/video streams and sensors information for remote navigation and control of the robot by non-expert users.

III. Hardware Architecture

MILO has modular hardware architecture and is built mostly from off-the-shelf components with the goal of developing a flexible and reliable platform. The main building blocks of MILO's hardware architecture are: The PC computer board, the hardware control board, and the base.

A. PC computer board

The PC computer board is responsible for local computing services, sending high-level control commands to the hardware control board, communicating the robot's status with the remote or local user, and MILO's connectivity to LAN/Internet.

MILO's PC computer board is a VIA Mini-ITX, 600 MHz x86 architecture single board PC consisting of, 20GB Notebook HD (expandable), 256MB of RAM (expandable), four USB, two FireWire, two RS-232 serial communication ports and one PCI expansion slot; see figure 2. The accessories are a USB 802.11 wireless module for network connectivity, Electret Microphone, 1/4" CMOS 300k pixels web-cam, and two speakers.

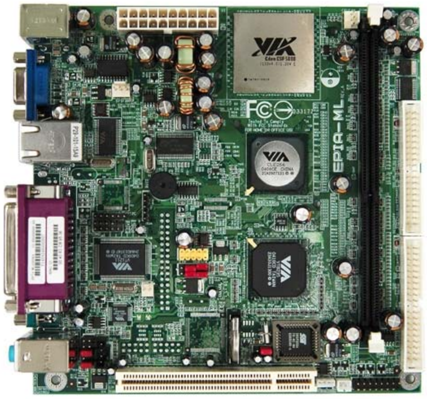


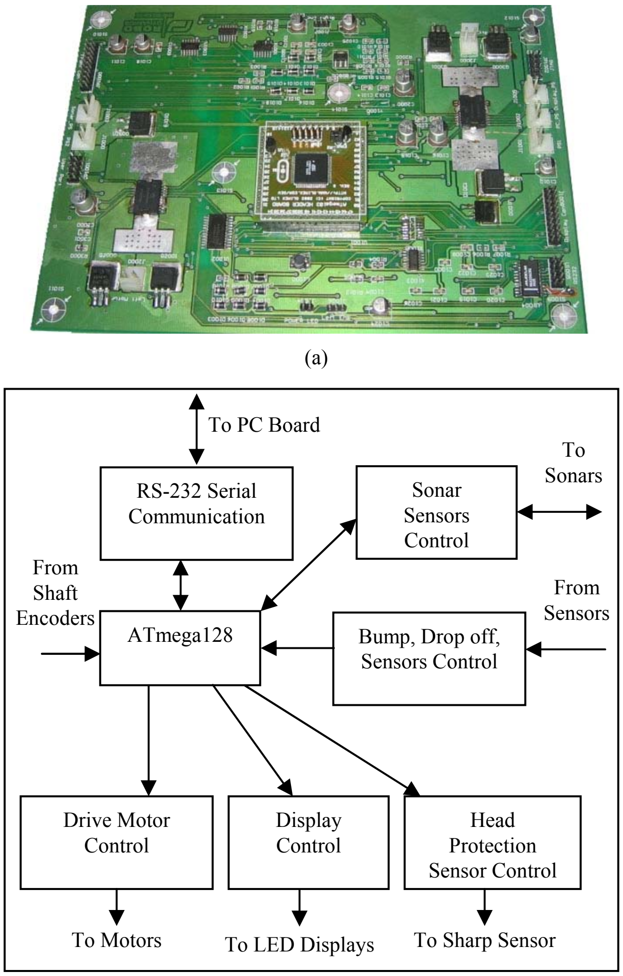
Fig. 2 MILO's PC Computer Board.

B. Hardware Control Board

The hardware control board is responsible for MILO's actuation, display and sensor control. This board processes the sensory information locally to autonomously avoid obstacles and communicates the relevant information with the PC computer board through a RS-232 serial bus; see figure 3a.

The hardware control board is based on an ATmega128 [15] microcontroller, which is a low power AVR 8-bit processor with 128 Kbytes of flash program memory, 4 Kbytes of EEPROM and 4 Kbytes of internal SRAM. The ATmega128 also includes an 8-channel 10-bit ADC, three timers, and several bus interfaces including two USARTs.

Specifically, the hardware control board consists of the following building blocks; see figure 3b:


(a)

(b)

Fig. 3 a) MILO's Hardware Control Board. b) Block diagram.

1) Drive Motors Controller: For each of the two drive motors an MC33486 chip, which is an automotive industry standard H-bridge with 10 Amps continuous current capacity has been used. It allows the motor to move forward and backward at a variable speed by changing the duty cycle of an input PWM signal.

2) Shaft Encoders Control: This module receives feedback from two shaft encoders in the form of digital pulses. This information is used to calculate the speed of each drive motor. Each shaft encoder is coupled with the shaft of a drive motor and generated 1024 pulses for every revolution.

3) Display Control: This module regularly updates two 8x8 LED grids serving as the eyes in the head of the robot displaying user-defined graphical patterns; see figure 4. This module consists of two 8-bits latches to scan rows and columns of each display fast enough to avoid flickering.

4) Head Position Control: This module generates PWM pulses to control an RC servo that moves the head of the robot up and down (pitch) about 90 degrees; see figure 4.

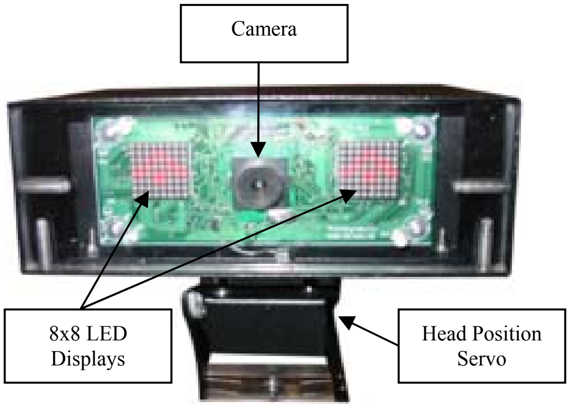
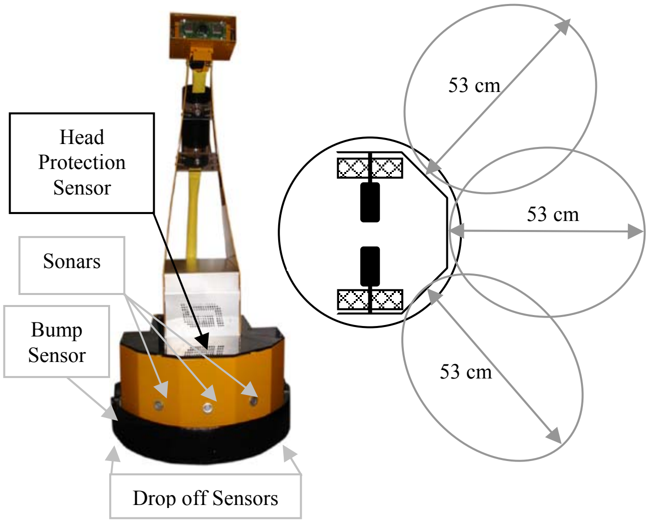


Fig. 4 MILO's Camera, two 8x8 LED Displays and the Head Position Servo

5) Sonar Sensors Control: This module drives three sonar sensors that are used to measure the distance of the robot from objects in front of and beside it Figure 5b shows the top view of the sonars coverage in front of and beside robot. The microcontroller continuously sends a pulse to each of the sonar sensors in order to generate an ultrasonic wave. The distance is proportional to the time distance between the generated pulse and the returned reflection; see figure 5a.

6) Bump and Drop off Sensors Control: this module is responsible for debouncing the digital inputs from the bump switches and the reflection based Infra-Red drop off sensors underneath the robot and interrupting the microcontroller; see figure 5a.


(a)

(b)

Fig. 5 a) MILO's Sonars, Head Protection Sensors, Bump Sensor and Drop off Sensors. b) Sonars coverage, top view.

7) Head Protection Sensor Control: This module is responsible for interfacing a Sharp GP2D120 IR Sensor to the microcontroller. This sensor is installed on the lower part of the body and is facing up. It protects the head and upper body of MILO from tall objects with a hollow bottom such as table tops; see figure 5a.

8) Serial Interface: this module is responsible for a two-way RS232 serial communication with the PC board at 56K baud rate; see figure 3b.

C. Base

MILO's base consists of the body, drive motors, and power module. The body is made from aluminium and Lexan, which is 105 cm high and weighs approximately 27 kg. The lower part of the body has a 52 cm diameter, which creates the housing for the motors, electronics, and battery. Two gear head motors with their 20 cm wheels plus two castors in the rear provide a differential steering mechanism capable of performing in place turns..

The power module includes a 12V 26Ah Sealed Lead Acid battery, which provides approximately 5 hours of continuous drive time. The battery, being an extremely heavy item is installed in the area between the motors and the castors first, in order to have enough support and second, to move the centre of mass toward the rear of the body.

IV. Software Architecture

MILO's software runs on the Windows XP operating system offering several advantages including: providing access to Microsoft Windows' available applications, technologies and infrastructure; the use of standard tools such as Microsoft's Visual Studio [16] for creation of 3rd party applications, and running applications concurrently onboard and off-board using Process Anywhere Technology™. The robot software consists of four main components; see figure 6:

A. Local Client

The local client is a Windows XP application that runs on the robot's internal PC computer board. It is responsible for allowing remote or local users to securely login to the robot and interact with it. The local client API provides the ability to develop 3rd party robotic applications by allowing access to information coming from the microcontroller, remote client and/or various attached devices such as USB, RS232, IEEE 1394, etc. and also to transmit messages back to those devices, the remote client or the microcontroller, all in an XML data format. Local client API uses Microsoft Visual Studios which can be used to create autonomous applications and can also launch and interact with any other Windows application. In addition the local client API can access the robot camera and microphone inputs via Direct X.

B. Remote Client

The Remote API provides the remote users with the ability to send/receive messages to/from MILO via LAN or Internet. All the communicated messages between remote and local clients are in XML format making the API OS Independent. Video and audio feeds that are sent by the local client over the network may be captured on the remote client via Direct X. In addition, the remote client can run concurrent local and/or remote applications using Process Anywhere Technology™. Also, any remote PC can potentially communicate with the remote client API using sockets. The remote client uses Microsoft Visual Studios to create remote applications including graphic control interfaces. Figure 7 shows a snapshot of the current version graphic control interface. This interface integrates the audio/visual streams and sensors status from the robot in a graphical form and can send motor control commands from keyboard or a joystick to the robot.

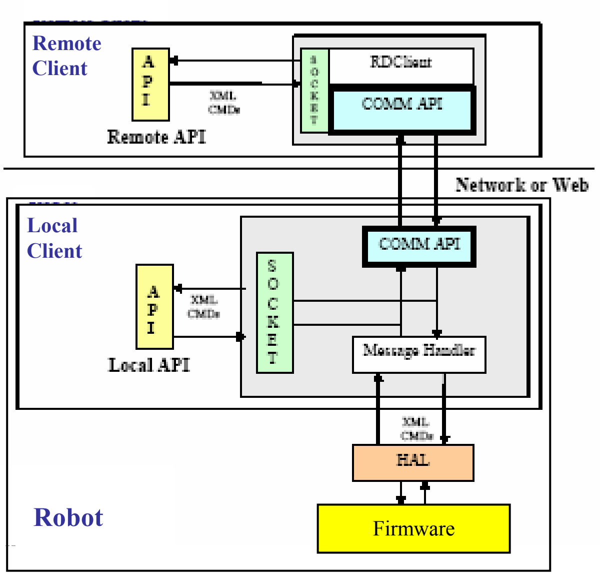


Fig. 6 MILO's Software Architecture

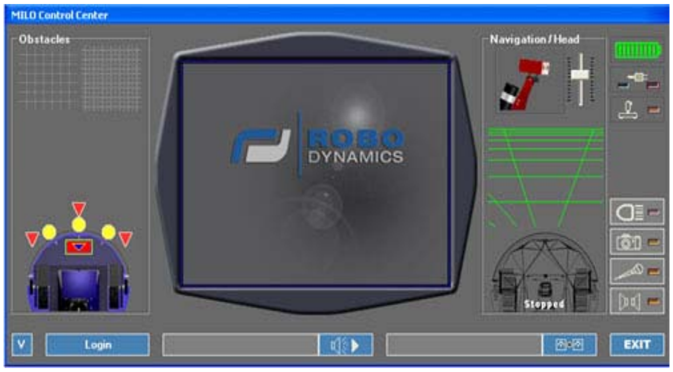


Fig. 7 MILO's graphic control interface snapshot

C. *Hardware Abstraction Layer*

The Hardware Abstraction Layer (HAL) is a separate Windows based module that is responsible for translating the messages between the local client and Micro Controller Unit (MCU). Using HAL makes the entire control system hardware independent. In order to support a new hardware, only HAL needs to be replaced. The messages that are sent to HAL are in XML format providing extensibility.

D. *Firmware*

Firmware is an interrupt-driven software module developed for the ATmega128 microcontroller in GNU C language. The firmware is responsible for two things: First, it communicates with the HAL module to receive control commands and to transmit current robot status to the user; and second, it autonomously stops the robot if there is an obstacle in front of or beside it (instincts). The latter task is performed independently of the commands that are being received from HAL and overrides them. This is necessary to ensure safety in several situations such as when there is a considerable amount of delay in the network resulting in the user's commands not reaching the robot on time; when the user, due to a limited field of vision, makes a mistake; or when the local client and/or HAL modules crash. This safety mechanism can be temporarily disabled by the user by sending a command to allow for the release of the robot from a dead-lock situation. Firmware consists of the following modules:

1) Interrupt Service Routine: this routine records the received interruptions from sensors in a queue and assigns priority to them.

2) PID controller: This controller corrects and synchronizes the speed of both drive motors by sending PWM pulses based on the feedback that it receives from the shaft encoders. This ensures the robot will travel on a straight line at a pre-specified speed when moving forward and enables accurate turns.

3) Sensors' Controller: This module controls all the available sensors on the robots. It measures the distance between the robot and obstacles beside it and in front of it by generating and sending 40 KHz pulses to sonar sensors and measuring the time interval between the generated pulse and its echo. In addition, the sensors' controller monitors the bump, drop-off, and head protection sensors for new events.

4) Communication & Execution Controller: This module establishes the MCU communication protocol with HAL. The protocol uses handshaking and checksum to establish an error free channel. It also interprets and executes the received control commands and generates status messages to be sent to HAL.

5) Autonomous Obstacle Avoidance: This module cooperates with the sensors' controller module to automatically stop the robot if it gets too close to an object or obstacle.

6) Head Position control: This open-loop control module generates PWM signals to control the head servo position.

7) Display Controller: This module generates bit streams to display received graphics patterns on two 8x8 LED matrixes and refreshes them periodically. This module is able to display sequences of graphics patterns, called movies.

V. EXPERIMENTS

Several experiments were conducted to evaluate the performance of MILO in a typical office environment where people and other objects are present. These experiments were performed on a flat carpeted floor with fully charged batteries.

In order to test MILO's differential steering mechanism, we remotely logged in to MILO from a wireless LAN and aligned MILO with a straight line drawn on the floor. Then, using a joystick connected to the remote host we sent a "move forward" command to MILO. After advancing 10 meters, MILO had deviated from the straight line. Upon repeating the experiment ten more times, we found the average deviation to be less than 1 cm, showing a 0.1% error.

Similarly, to test MILO's turn-in-place ability, a circle was drawn on the floor around the lower part of MILO's base. The "turn-in-place" command was then sent. After rotating ten times, the average deviation was less than 1 cm which, relative to MILO's 50 cm diameter, shows a 2% error.

An 8-bit register determines MILO's speed zero value as the stopped position. MILO starts moving at the value of 50 with the speed of 12 cm/s and reaches the maximum speed of 25 cm/s at 255. The motors' maximum speed however, is higher than the preset values, but for safety reasons this maximum is not used.

In another series of experiments MILO was remotely navigated around the same office, this time going through doors and being parked in a small spot using the remote graphical control interface. The interface received a video stream from a web-cam mounted in the robot head and the commands were issued from a joystick. The user was also able to see the sonar readings as well as the bump and drop-off sensor values. This experiment was performed from the following locations with different bandwidth and delays: 1) A LAN in the office. In this case, the user was not able to see the robot. 2) From a remote high-speed site in town. 3) From a home with DSL connection in town. 4) From homes abroad (Canada and Spain) with DSL lines. In all of these cases, users were able to navigate the robot successfully and perform the requested tasks. In cases 3 and 4 which had the most and varying delays, the autonomous obstacle avoidance system prevented the robot from running into objects.

Finally, we navigated MILO up and down an industrial ramp (15°). In both the ascent and descent, MILO possessed enough torque to perform the task with ease. In addition, when MILO was stopped on the ramp, despite its weight, the high gear ratio of the motors prevented it from rolling downwards.

VI. Conclusion and Future Work

A new general purpose and low cost mobile platform called MILO was presented. It was discussed how and why MILO is suitable for a wide array of personal applications such as commercial security, mobile teleconferencing, tele-presence, personal assistance, elderly care and robotic research. Low cost, PC architecture, expandability, its easy accessibility, form factor as well as its successful remote navigation were some of the components that make MILO an effective robotic platform.

In the future, we plan to use MILO to further develop applications for security, tele-presence, and personal assistance. These applications will require new software applications, integration of new sensors such as heat sensors, and higher resolution cameras. Other plans also include the incorporation of autonomous self-charging, a necessity for security applications. Additionally, controlling multiple MILO robots from one remote control user interface, autonomous navigation, voice interface, and vision systems for recognizing faces and objects are other areas of future research and development. These additions will give MILO the necessary edge and affinity for future applications in the robotics forum.

Acknowledgments

We would like to thank Sharon Haughey and Sepi Chitsaz for their assistance in editing the text and providing images.

References

[1] Nilsson, Nils J. *Shakey The Robot*, Technical Note 323. AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, Apr 1984

[2] Sara Kiesler, Pamela Hinds "Human-Robot Interaction". Special Issue of *Human-Computer Interaction*, Volume 19 (2004), Numbers 1 & 2.

[3] H. Choset and K. Nagatani. Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2):125–137, Apr. 2001.

[4] Bergbreiter S., Pister K.S.J, "CotsBots: An Off-the-Shelf Platform for Distributed Robotics", IROS, Las Vegas, Nevada, October 2003

[5] P. Will, A. Castaño and W.-M. Shen, ``Robot Modularity for Self-Reconfiguration" Proc. SPIE Sensor Fusion and Decentralized Control II, pp. 236-245, Boston, Sep. 1999.

[6] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, NY, 1999.

[7] http://marsrovers.jpl.nasa.gov/home/index.html

[8] http://www.newscientist.com/article.ns?id=dn3661

[9] http://gen.surg.uic.edu/misc/robot.htm

[10] http://www.irobot.com/consumer/product_detail.cfm?prodid=15

[11] http://www.activrobots.com/

[12] http://www.evolution.com/er1/

[13] http://www.sonystyle.com/is-bin/INTERSHOP.enfinity/eCS/Store/en/-/USD/SY_BrowseCatalog-Start?CategoryName=AIBO

[14] http://www.robosapienonline.com/

[15] http://www.atmel.com/dyn/products/product_card.asp?part_id=2018

[16] http://msdn.microsoft.com/vstudio/