Honors Undergraduate Research: Autonomous Robot for Remote Detection of UXO

Joshua Galloway[1], Daren Wilcox[2]

Abstract – The Electrical and Computer Engineering Technology (ECET) Honors student developed a prototype for a landmine and unexploded ordinance (UXO) seeking robot with the objective of achieving fully autonomy. The system provided functionality including: locating metallic landmines and UXO within a defined area/environment, recording the location of said landmines and UXO's, and storing the data off unit via an IEEE 802.11b/g connection to a Windows or Linux-based laptop computer. Although fully autonomy was not achieved, application of the prototype and corresponding research may lend themselves to de-mining the more than 100 landmine/unexploded ordinance affected countries in the world particularly in desert terrain (US Department of State Fact Sheet, 2 July 2003).

Keywords: Robotics, UXO, de-mining, remote detection, honors

INTRODUCTION

The United Nations estimates that 2,000 people are killed or maimed by mine explosions each month, and for every mine cleared, twenty more mines are laid. Of the people affected a vast majority are civilians[1]. Of the mines removed, most are removed by trained personnel wearing ballistic armor and probing the ground with a baton, which puts the personnel within an unsafe distance from the explosive. This task is made less dangerous through the application of robotics and other technologies. However, these technologies are of considerable expense, and since most of the mines and UXO are in the global south, expense is a great concern[2]. The goal of the research preformed by the student was to arrive at a prototype that could autonomously sweep a field and record the location of any metal detected. Because of diverse locations in which landmines have been deployed, a single terrain was selected based on the ubiquity of landmine and UXO deployment in each affected environment. The robot was designed around this selection.

SYSTEM LEVEL DESIGN

The student's robot, in order to contend with the problems of cost and effectiveness in available solutions, should be a system that is inexpensive and performs a portion of the hand prodding process. With the problem of cost in mind, it was decided that to produce a robot that would be able to locomote in all the varied terrains that are affected by landmines and UXO's was not viable. Therefore, the robot was designed for application in a specific region to keep cost at a minimum. Also, any data the robot acquired should be stored off-unit due to the possibility of destruction of the robot.

Figure 1: Robot Top and Side View

1 Southern Polytechnic State University, 1100 S. Marietta Parkway, Marietta, GA 30060-2896, jgallow2@spsu.edu

2 Southern Polytechnic State University, 1100 S. Marietta Parkway, Marietta, GA 30066-2896, dwilcox@spsu.edu

Locomotion

Terrain Selection. In order to select a locomotion system for the robot, it is necessary to decide on the terrain it must negotiate. The environment that the robot is tailored for should be one of the most landmine-affected environments. Using the countries from Table 1, the types of terrain present in each country was tabulated.

Table 1: Countries Affected by Landmines[3]

County	Estimated Number of Landmines
Egypt	23,000,000
Iran	16,000,000
Angola	15,000,000
Afghanistan	10,000,000
China	10,000,000
Iraq	10,000,000
Cambodia	6,000,000
Vietnam	3,500,000

Table 2: Type of Terrain in Affected Countries

Country	Type of Terrain			
	Sand	Grassy Field	Rice Paddy	Rocky/Rough
Egypt	X			
Iran	X			X
Angola		X		X
Afghanistan	X	X		X
China	X	X	X	X
Iraq	X			
Cambodia		X	X	X
Vietnam		X	X	X

Taking the information from Table 2 and weighting each "X" in an ad hoc method, a numerical score for each type of terrain was derived. Each country's weight-constant, C_N, was arrived at by dividing its total number of landmines by the country with the largest total landmines.

C_N = *(Estimated # of landmines in the country)/23,000,000*

The score for each terrain was derived summing the total weights of the countries with and "X" in a particular terrain's column.

$Score = \sum C_N * X_N$

Here is an example calculation.

$Score_{Rice\ Paddy} = C_{China} + C_{Cambodia} + C_{Vietnam} = 0.435 + 0.261 + 0.152 = 0.848$

Table 3: Terrain Scores

	Sand	Grassy Field	Rice Paddy	Rocky/Rough
Adjusted Score	3.000	1.935	0.848	2.630

As a result of the tabulation in Table 3, sandy terrain will be the medium in which the robot will have the greatest impact. Since Egypt is geographically in keeping with this choice and also has the most total landmines of any country, the types of landmines in Egypt were researched to find what the robot would need to be able to detect. It was found that the Western Desert of Egypt, the area with the largest problem, is populated with mines from the

World War II era. In addition, the majority of the types of mines deployed have sufficient amounts of ferrous metal in them as to be detected by a simple metal detector[4].

Locomotion System Selection. Three types of systems were considered: a two-wheeled system with a support, a four-wheeled system, and a pedal system.

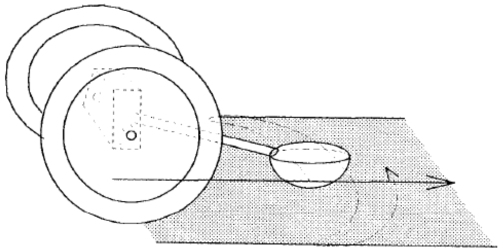

Figure 2: Two-Wheeled System with Support[5]

The two-wheeled system has the advantages of being simple in design, very maneuverable, and inexpensive. The system was found wanting in its possible applications because the support may become stuck in soft sand and certainly would not be applicable to even moderately rough terrain.

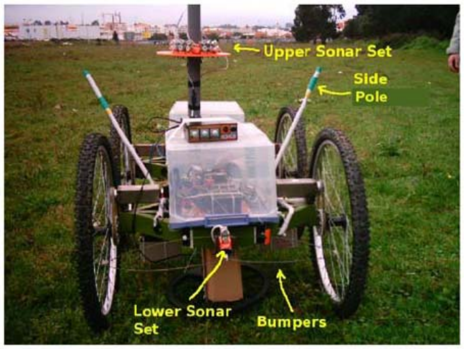

Figure 3: Four-Wheeled System[6]

The four-wheeled system is a very solid system, and unlike the two-wheeled system, it would not be hindered by small pot-holes or slightly rough terrain. These attributes will make the system easily portable to other environments. The only striking disadvantage is that the system's turning radius may be comparatively large, which could hinder maneuverability.

Figure 4: Pedal System[7]

The pedal system is exceedingly maneuverable and can negotiate irregular terrain well. However, this comes with a large increase in complexity and, therefore, expense. The system would be very difficult to implement and may result in a slow-moving system. The additional motors needed to create the dexterous movements necessary for any selected gate will result in a relatively large power consumption.

The systems were all rated numerically from one to five (one being poor, and five being excellent) for the expected system mobility and performance in each of the four types of terrain.

Table 4: Locomotion Systems Rating

Type of System	Types of Terrain			
	Sand	Grassy Field	Rice Paddy	Rocky/Rough
Two-Wheeled with Support	3	4	1	1
Four-Wheeled	4	4	3	3
Pedal	2	3	4	4

From Table 4 and the types of landmines and UXO's found in the selected territory of the Western Desert of Egypt, it is clear that the system should be of the four-wheeled type. The four-wheeled system is easily implemented, maneuverable enough for a desert environment, and low in power consumption.

Locomotion Implementation. To rapidly and inexpensively create a working robot, the student used the chassis, motors, and H-bridge of a radio controlled (RC) car. A New Bright 1:8 Radio Control Full Function 4 Door Jeep Wrangler Unlimited was selected as the full implementation of the locomotion system based on size and cost. The decorative parts were removed and the car's electronics were tested in order to find the front and rear H-bridge's control lines.

Control System

Control System Selection

Because of the possibility that the robot may be destroyed, it was decided to keep as much of the system expense off-unit. The scheme settled upon was to control the robot via IEEE 802.11b/g wireless local area network standards to link a small microcontroller to a personal computer. This allows the system to be composed of readily available networking technology and puts only an inexpensive microcontroller and wireless router on the unit. Also, destruction of the robot during a sweep will not result in a loss of previously acquired data, as it will be stored in the laptop. A block diagram of the proposed system is shown in Figure 5.

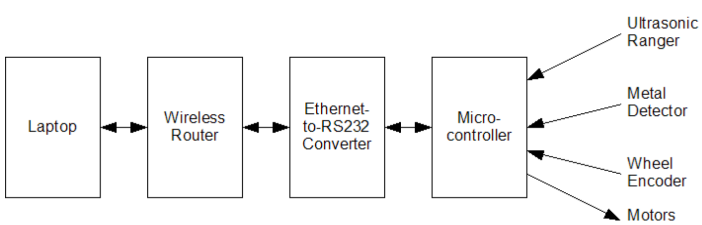

Figure 5: Control Block Diagram

Control System Implementation. The various parts of the control system were implemented as follows.

- Laptop—This was implemented with an Acer Aspire 1640 IEEE 802.11b/g enabled laptop computer with Windows XP and Ubuntu 8.10 installed in a dual-boot setup. However, any computer that can support an accessory to provide the necessary IEEE 802.11 compliance, and load Java's runtime environment will be sufficient. The target system would be Asus Eee PC or similar low cost laptop. Cost: $200-$250.
- Wireless Router—This was implemented with a Linksys WRT54GTL IEEE 802.11b/g router. Its firm ware was upgraded to the newest version available. Cost: $60.

- Ethernet-to-RS232 Converter—A Maxport ethernet-to-RS232 converter available from Comfiletech was used to realize this portion of the control scheme. Again, the firmware was upgraded to the newest available version upon receipt. Cost: $75.
- Microcontroller—The PIC18F4520 was the microcontroller selected for the robot's input-output (IO) functions and to interface the Maxport. It was selected because it provided many of the functions necessary to interface the robot's sensors, which allowed for a smaller control-logic footprint on the unit. Cost: $4.

Sensors

Sensor Selection. To keep the cost and power consumption of the unit low, the number of sensors was kept to a minimum. The robot needed to detect the ferrous metal present in the targeted, World War II-era landmines and UXO, which means a metal detector was needed. Also, the robot needed a way to detect objects in its path, so an ultrasonic ranger was necessary. In addition to the ultrasonic ranger, a web-camera was added to allow the unit to be teleoperated and possibly take pictures of the offending areas during an autonomous sweep. Finally, the robot must be able to map its exact location relative to an arbitrarily selected point. This point was selected as the starting point of the sweep, and odometry was selected as the means of mapping its location.

Sensor Implementation. The sensors were implemented as follows.

- Metal Detector—A Slinky Treasure Tracker Metal Detector was eventually settled upon based on cost and speed of implementation. The unit was probed until a suitable control signal was found, and then, it was interfaced to the microcontroller. Cost $30.
- Ultrasonic Ranger—MaxBotix's LV-MaxSonar-EZ4 was used to implement this sensor. The ranger was reputed to be able to detect objects from 6 inches out to 254 inches. The beam width is 2 feet, and the refresh rate is 20 Hz. The selected interface to this device is analog. Cost: $30.
- Web-Camera—The web-camera selected was the Trendnet TV-IP100 RJ45 Internet Camera Server. The selected camera was lowest in cost of the systems available, and could be integrated by simply providing power and connecting it to the router. Only a web-browser is required to interface the device. Cost: $80.
- Odometry—A CTS 288 Series 4-Bit Gray Code Rotary Encoder was decided upon to implement the robot's odometry system. It was coupled to one of the rear-drive wheels via a 2.5 inch diameter hobby airplane tire. Cost: $10.

Power Distribution

Power System Selection. In order to keep costs low, the decision was made to utilize the RC car's provided 9.6 V nickel cadmium (NiCd) battery for the control and sensor systems with the exception of the metal detector, which was powered via a 9 V PP3 battery. The RC car's motors and H-bridges were powered by way of a separate 9.6 V NiCd battery in order to isolate the control logic from the noise created by the motors. It was necessary to generate several voltages from nominal 9.6 V batteries in order to power the system's varied devices.

Power System Implementation. The power system was implemented thusly.

- Control Logic Battery—This was the 9.6 V battery that was provided with the New Bright 1:8 Radio Control Full Function 4 Door Jeep Wrangler Unlimited used for the locomotion system.
- Motor Battery—A 9.6 V Black & Decker Model No. GC9601SB cordless drill was disassembled for its NiCd battery and charging system. This was, again, done to curb cost, as the drill, with its charger included, was far less expensive than a traditional NiCd battery and charger.
- Metal Detector—This was powered via a PP3 9 V battery.
- Microcontroller and Remaining Sensors—This portion of the robot was powered though the control-logic battery and a LM7805 linear 5 V, 1 A regular. This portion of the system drew a total of 250 mA at 5 V.

- Linksys Router—This was powered by way of the 9.6v control-logic battery and a LM2577-12 step-up switching regulator. The router drew 500 mA at 12 V.
- Trendnet Web-Camera—The camera drew 2.5 A at 5 V, so it was necessary for the device to be put on its own 5 V regulator powered through the control-logic battery. A 5 V, 3 A, linear, LM323 regulator was used to provide the requisite voltage and current.
- Drive Motors and H-Bridge—These devices were all powered by way of the motor battery; no regulation was needed.

SOFTWARE

The software for the system was broken into two main parts: the PIC18F4520 program and the graphical user interface (GUI)/control software for the laptop. The PIC1F4520 software was written in microchip's proprietary PIC18 series assembler, and the laptop or pc software was written in Java. The use of Java makes the program inherently portable to other operating systems besides Windows—such as Linux.

PIC18F4520 Software

The PIC's program performed three main functions: interfacing the sensors, relaying and receiving data from the pc, and controlling the robot's locomotion.

Executing a Move Command. In order to maintain accurate results from the odometry system, slippage of the wheels had to be minimized. This was done by applying an approximately parabolic increase in average voltage to the drive motor unit until the robot begins to move.

Main PIC18 Program. The main program for the PIC18F4520 utilizes a polling procedure to wait for the pc to send a command. Upon receipt of the command, the system executes it, updates its sensor registers, and sends various status packets and acknowledgements.

Java Program

The Java program used to control the robot provides a dual functionality. It performs the proposed autonomous sweeping action while storing all metal detection hits to a text file, "autonomous_sweep.txt," and allows the user to manually guide the robot. The program was compiled with version 1.4.2_16 of Java2's software development kit, and it was tested with Java runtime environment j2re1.4.2_16.

Manual Mode

The controls move the robot in the direction of the arrows a prescribed distance, which is entered in the text box below the controls, and updates the status window as each move and status packet is sent and received. The status button performs a status ping to the robot and returns the value of all the status registers in the PIC18f4520, which is then printed to the status window. The view camera button opens the computer's default internet-browser to the IP address in the appropriate box. It also records the time the browser was opened in the status window. In fact, all commands in manual mode are time-stamped. Through the functionality provided by the manual mode, the robot may be used as a teleoperated metal detector with visual feedback from the robot's location.

Autonomous Mode

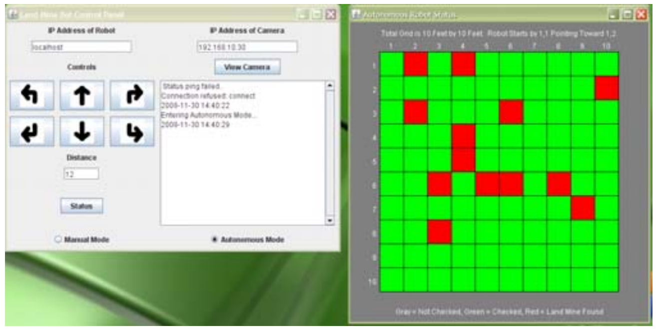

Figure 6: Autonomous Mode

The autonomous mode performs an autonomous sweep across a predefined 10 foot by 10 foot area. The system provides a progress or status map that shows an approximate location of the robot and an approximate location of any ferrous metal located during the sweep. The sweep is executed on its own thread to keep from locking the GUI interfaces. The control panel is present during the sweep, but its buttons are locked to keep from affecting the autonomous sweep. When a ferrous metal is detected, the program records the metal's location in the form of x and y coordinates in a text file called "autonomous_sweep.txt." This allows the system's user to locate any detected metal within the full available accuracy of the system.

COMMUNICATIONS PROTOCOL

The system, as a whole, communicates over the IEEE 802.11b/g protocol. However, a custom command set was created to interface the Java and PIC programs. This protocol included a two-byte packet for commands sent to the robot and a three-byte status message as well as a one-byte move-command-received acknowledgement from the PIC to the pc.

Communication to the PIC18F4520

The packet received by the PIC18F4520 is composed of two bytes. The high byte (the first byte received) has two pieces of data in it: the command type, and distance to travel. In the case of a status ping, the most significant bit (MSB) of the high byte will be set; if not, the command is a move command. The remaining seven bits contain the distance that the robot is being prompted to move if the MSB is not set. The low byte contains the direction in which the robot should move if the MSB is set. The following graphic shows the data contained in the packet.

1-bit	7-bits	8-bits
S/!M	Distance	Direction

Figure 7: Command Packet

- S/!M (1-bit)—Command Type. 1 = Request status packet, 0 = Move Command
- Distance (7-bits)—The distance the robot is to move.
- Direction (8-bits)—The direction in which the robot is to move. This byte-sized value is the bit configuration that must be inclusively ORed with a cleared PORTC of the PIC18F4520 to produce the requested directional movement. In Table 5, the bit configurations and their resulting direction are shown.

Table 5: Directional Commands

Hexadecimal Low Byte Value	Resulting Movement
0x18	Forward Left
0x10	Forward
0x11	Forward Right
0x28	Backward Left
0x20	Backward
0x21	Backward Right

Communication to the PC

The PIC18F4520 response to a status pin by sending a three-byte packet to the computer, which contains the data from its freshly updated status registers. In like fashion, a move command also completes by sending data from the PIC's status registers; however, a one-byte move acknowledgement is sent prior to the execution of the command. The move acknowledgment is 0x0F, and is sent to show that the command was received. The three-byte status packet contains information on how far from the beginning of the last move command execution was metal detected, if there was metal detected during the last move, the ranger output, and the total distance traveled during the last move command. Figure 8 is a visual of the packet's payload.

8-bits	1-bit	7-bits	8-bits
Distance to Metal	M	Range	Distance Traveled

Figure 8: Status Packet

- Distance to Metal (8-bits)—This the distance form the beginning of the last executed move command to the any detected metal.
- M (1-bit)—Metal found in last move. 1 = yes, 0 = no.
- Range (7-bits)—The output of the ultrasonic ranger.
- Distance traveled (8-bits)—The distance actually traveled during the last executed move command.

RESULTS

The robot was tested in manual mode at a nearby baseball field.   The system was run through a ten foot by ten foot grid with landmine simulates, in the form of half-inch galvanized washers, scattered randomly inside the grid. Figure 13 shows the grid on which the robot was tested.

The robot was started at the point marked 0,0 of Figure 10. The sweep was performed by moving in a forward motion from 0,0 to 0,9, repositioning the robot to the next row at square 1,9, and then, sweeping in a backward motion (the robot moving in reverse) to 1,0. This process was repeated until the field had been completely covered. Figure 9 shows a screen-shot of the sweep in progress.

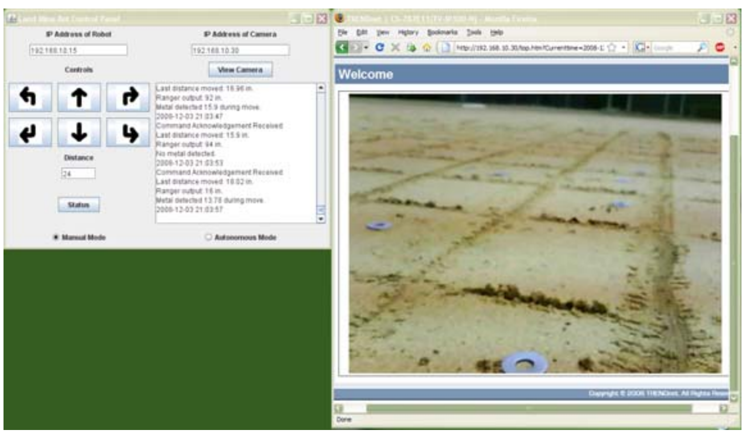

Figure 9: Sweep in Progress

The simulates were laid out as is depicted in Figure 10.

	0	1	2	3	4	5	6	7	8	9
0		X								
1			X						X	
2		X								
3	X			X			X			
4		X				X		X		
5	X					X	X		X	
6			X					X		X
7	X				X			X		
8		X		X				X		X
9	X		X	X	X				X	

Figure 10: Simulate Locations

The robot completed the sweep in ten minutes. This is less the time it took to recharge the battery on the Acer Aspire 1640 series laptop, which ran out when the robot was on square 3,2 of the matrix. The sweep results are shown in Table 6. Figure 11 shows the spot as which the robot was when the laptop battery failed. The sweep was continued after recharging the battery, and the robot was not moved in the interim.

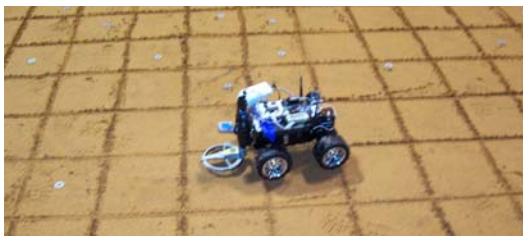

Figure 11: Robot after Laptop Battery Failed

Table 6: Sweep Results

Total Simulates	30
Total Successful Detections	27
Total Missed Detections	3
Total False Detections	0
Probability of Detection	90%

The missed detections were in squares 3,3; 5,6; and 9,3. Both the simulates in 3,3 and 5,6 were not run over by the metal detector. The simulate in 9,3 was run over but the detection was recorded in the same move sequence as the correctly detected simulate in square 9,4. So, the detection was effectively not recorded.

CONCLUSION

The robot was tested in manual mode with a high degree of success. The majority of the missed landmine simulates were missed because they did not pass under the robot's metal detector. This could be rectified by increasing the size of the metal detector or by decreasing the width of the rows in the search pattern. No false detections occurred during the test, and the robot never lost communication with the controlling laptop except when the laptop battery failed. This shows that the robot could easily function as a teleoperated metal detector in its current state, and it could be used to magnetically map an affected area from a safe distance.

The network portion of the Java program functioned as was intended. The system was able to communicate bi-directionally over a wireless network, and the robot performed move commands with a relatively high degree of accuracy. The autonomous portion of the Java program did not perform consistently enough in simulation to allow for feasibility of a real-world test. The simulated robot and controlling program intermittently became hung when reaching the end of a search row.

The Java program development hindered the testing of the autonomous operation of the robot. The semester ended before the programming bugs could be resolved. It is expected that further investigation into the odometry requirements would be necessary.

The router and wireless network functioned well under what would have been the intended circumstances. With no interference from surrounding wireless networks, the system functioned without any drop outs for the duration of the control-logic battery's charge. Stray radio frequencies (RF) in the 2.4 GHz range would not be an issue in a remote mine field. However, near the university—where the preliminary testing was performed—stray RF caused the robot to routinely lose connection with the laptop.

Several of the robot's subsystems functioned very well. The step-up regulator constructed to power the 12V router off of a 9.6V battery worked as did all the regulators. The LCD was properly controlled by the PIC18F4520, and the ultrasonic ranger produced accurate results under most circumstances. The only failing of the ultrasonic ranger was in the student's selection of the model with the narrowest detection beam. A narrow beam is, according to the company's applications engineer, prone to malfunction when an object is in the beam's periphery.

Overall, the system showed it could perform the function of sweeping a mine field under manual control, and given more time than one semester, the system could perhaps perform this function autonomously. For $250 and, if necessary, the price of a laptop, the robot could provide a landmine removal team with some idea of what awaits them in a suspected minefield without endangering any personnel. The system provides proof of concept.

ACKNOWLEDGEMENTS

Special thanks to the Electrical and Computer Engineering Technology Department Heads Association (ECETDHA) of the American Society of Engineering Education (ASEE) and its Chair Ronald Land for their financial support of this research through the endowment of a $1000 mini-grant.

REFERENCES

[1] United Nations, "Landmine Factsheet" [Online Document], 1997 Sept., [cited 2008 Nov. 22], Available: http://www.un.org/cyberschoolbus/banmines/facts.asp

[2] United Nations, "Countries Affected by Landmines", [Online Document], 1997 Nov., [cited 2008 Nov. 22], Available: http://www.un.org/cyberschoolbus/banmines/resources/affected.asp

[3] United Nations, "Countries Affected by Landmines", [Online Document], 1997 Nov., [cited 2008 Nov. 22], Available: http://www.un.org/cyberschoolbus/banmines/resources/affected.asp

[4] United Nations Mine Action Service, "Mine Action Assessment Mission Report", [Online Document], 2000, [cited 2008 Nov. 22], Available: http://www.mineaction.org/downloads/Egypt_Assessm_Report.PDF

[5] Nicoud, J.D. and M. K. Habib, "The Pemex-B Autonomous Demining Robot: Perception and Navigation Strategies", [Online Document], 1995 Aug., [cited 2008 Nov. 22], Available: http://ieeexplore.ieee.org/iel3/3952/11432/00525830.pdf?isnumber=11432&prod=CNF&arnumber=525830&arSt=419&ared=424+vol.1&arAuthor=Nicoud%2C+J.D.%3B+Habib%2C+M.K.

[6] Santana, Barata, Correia, "Sustainable Robots for Humanitarian Demining", [Online Document], 2007, [cited 2008 Nov. 22], Available: http://www.ars-journal.com/International-Journal-of-Advanced-Robotic-Systems/Volume-4/ISSN-1729-8806-4205.pdf

[7] DeTeC, "Demining Robots, Anti-Personnel Mine Removal", [Online Document], 1997 Nov., [cited 2008 Nov. 22], Available: http://diwww.epfl.ch/lami/detec/rodemine.html