DESIGN OF A NON-INVASIVE EMERGENCY RESPONSE TEAM ROBOT

A Thesis
Presented
To the Faculty of
California State University, Chico

In Partial Fulfillment
of the Requirements for the Degree
Master of Arts
in
Computer Science

by

© Angie F. Shia 2006

Spring 2006
DESIGN OF A NON-INVASIVE EMERGENCY RESPONSE TEAM ROBOT

A Thesis

by

Angie F. Shia

Spring 2006

APPROVED BY THE INTERIM DEAN OF THE SCHOOL OF GRADUATE, INTERNATIONAL, AND INTERDISCIPLINARY STUDIES:

______________________________
Susan E. Place, Ph.D.

APPROVED BY THE GRADUATE ADVISORY COMMITTEE:

______________________________
Anne Keuneke, Ph.D., Chair

______________________________
Benjoe Juliano, Ph.D.
PUBLICATION RIGHTS

No portion of this thesis may be reprinted or reproduced in any manner unacceptable to the usual copyright restrictions without the written permission of the author.
DEDICATION

This thesis is dedicated to my loved ones that made sacrifices to allow this thesis and my graduate school to happen: To J.C, I could have never made it without you. To Simon D., if only I knew; words cannot describe my regrets. To Adam and Eve D., I wished there were more time. To Tony, Ling Ling and the rest of the gang, thank you all for your understanding and support.
Search and rescue robotics is an area that challenges the very depth and breadth of us as human beings and computer scientists. It involves many different specialties and studies, from medicine, to science, to psychology. It requires us to be always on our toes and to think literally in the box, as many victims and rescue robots are in, literally, an environment the size of a shoebox. The technical difficulties involved in search and rescue robots are numerous, but the reward is equally great. It is my hope that this thesis will make some contribution to the men, women, animals, and robots involved.
ACKNOWLEDGMENTS

I would like to first thank my graduate advisor, Dr Anne Keuneke, for her guidance, patience, and tremendous support. I also like to thank my committee member Dr Benjoe Juliano, for taking time out of his busy schedule to offer guidance. I also like to thank Tony for editing such a long document and Ling Ling for being there during a critical time.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Publication Rights</th>
<th>iii</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dedication</td>
<td>iv</td>
</tr>
<tr>
<td>Preface</td>
<td>v</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>vi</td>
</tr>
<tr>
<td>Abstract</td>
<td>xiii</td>
</tr>
</tbody>
</table>

## CHAPTER

### I. Introduction

- The Problem.................................................................................................................. 1
- Relevant Work Done...................................................................................................... 2
- Insight to Resolving the Problem
  - Difficulty in Breathing.......................................................................................... 3
  - Dehydration................................................................................................................ 3
- Bleeding......................................................................................................................... 3
- Medication..................................................................................................................... 4
- Oversight......................................................................................................................... 4
- Proposed Solution.......................................................................................................... 4
- Organization.................................................................................................................... 6

### II. Background Information

- Search and Rescue......................................................................................................... 7
- Role of Emergency Response Team................................................................................ 8
- First Aid for Confined Space......................................................................................... 9
- Overview of FEMA’s Urban Search and Rescue Agency
  and The Center For Robot-assisted Search and Rescue............................................. 11
- Introduction to Mobile Robots.................................................................................... 12
  - What is a Robot?........................................................................................................... 12
  - What is a Mobile Robot?............................................................................................. 12
  - Classification of Mobile Robots................................................................................. 13
<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autonomous Agents and Robots</td>
<td>14</td>
</tr>
<tr>
<td>The Need for Autonomy</td>
<td>15</td>
</tr>
<tr>
<td>Medical Diagnostic System and Rescue Robots</td>
<td>15</td>
</tr>
<tr>
<td>III. Related Research: Search and Rescue Robotics</td>
<td>17</td>
</tr>
<tr>
<td>Classification and Requirements for Search and Rescue Robots</td>
<td>17</td>
</tr>
<tr>
<td>Cart-and-Wheel Robots</td>
<td>20</td>
</tr>
<tr>
<td>Biology-inspired Robots</td>
<td>24</td>
</tr>
<tr>
<td>Aeronautical Robots</td>
<td>27</td>
</tr>
<tr>
<td>Issues with the Current Designs and Search and Rescue Paradigm</td>
<td>29</td>
</tr>
<tr>
<td>The Need for a Special Rescue Robot</td>
<td>31</td>
</tr>
<tr>
<td>What Do Victims Need?</td>
<td>31</td>
</tr>
<tr>
<td>What Can the Robot Carry?</td>
<td>32</td>
</tr>
<tr>
<td>Looking Ahead</td>
<td>36</td>
</tr>
<tr>
<td>IV. Emergency Response Team Robot Requirements and Specifications</td>
<td>37</td>
</tr>
<tr>
<td>Payload Requirements and Specifications</td>
<td>37</td>
</tr>
<tr>
<td>Feasibility versus Criticality</td>
<td>37</td>
</tr>
<tr>
<td>Other Issues Affecting Payload</td>
<td>41</td>
</tr>
<tr>
<td>Hardware Design Requirements and Proposal</td>
<td>42</td>
</tr>
<tr>
<td>Understanding Hardware Design Requirements</td>
<td>42</td>
</tr>
<tr>
<td>ERT Robot Design</td>
<td>46</td>
</tr>
<tr>
<td>Software Design Requirement</td>
<td>50</td>
</tr>
<tr>
<td>Software Considerations in ERT Robot Designs</td>
<td>50</td>
</tr>
<tr>
<td>AI in Robot Control Programs</td>
<td>53</td>
</tr>
<tr>
<td>Software Consideration for the ERT Robot</td>
<td>54</td>
</tr>
<tr>
<td>V. Module Design and Implementation</td>
<td>58</td>
</tr>
<tr>
<td>System Architecture and Design</td>
<td>58</td>
</tr>
<tr>
<td>Modules Overview</td>
<td>60</td>
</tr>
<tr>
<td>State Transitions</td>
<td>62</td>
</tr>
<tr>
<td>Locating Victims Using a Non-visual Method</td>
<td>65</td>
</tr>
<tr>
<td>Prototype Materials</td>
<td>66</td>
</tr>
<tr>
<td>Robot Cost</td>
<td>68</td>
</tr>
<tr>
<td>Issues with Implementing the Proposed Hardware Design</td>
<td>68</td>
</tr>
<tr>
<td>CHAPTER</td>
<td>PAGE</td>
</tr>
<tr>
<td>-----------------</td>
<td>------</td>
</tr>
<tr>
<td>Sample Interactive and Diagnostic Program</td>
<td>69</td>
</tr>
<tr>
<td>Communicating with Humans</td>
<td>69</td>
</tr>
<tr>
<td>Sample Algorithm and Reasoning Model</td>
<td>72</td>
</tr>
<tr>
<td>Sample Module Implementation</td>
<td>73</td>
</tr>
<tr>
<td>Collaborating with Other Robots</td>
<td>83</td>
</tr>
<tr>
<td>Communication Design Issues</td>
<td>83</td>
</tr>
<tr>
<td>VI. Conclusion</td>
<td>85</td>
</tr>
<tr>
<td>Review of Discussion</td>
<td>85</td>
</tr>
<tr>
<td>Main Points for the design of an Emergency Response Team Robot</td>
<td>86</td>
</tr>
<tr>
<td>Research Recommendation</td>
<td>86</td>
</tr>
<tr>
<td>Conclusion</td>
<td>88</td>
</tr>
<tr>
<td>References</td>
<td>89</td>
</tr>
<tr>
<td>TABLE</td>
<td>PAGE</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>1. Classification of Robots</td>
<td>14</td>
</tr>
<tr>
<td>2. Problems and Corresponding Remedy Chart</td>
<td>32</td>
</tr>
<tr>
<td>3. Minimum Space for Weight Requirements for a Basic Mobile Robot</td>
<td>39</td>
</tr>
<tr>
<td>4. Minimum Space and Weight Requirements for a Basic Search and Rescue Robot</td>
<td>40</td>
</tr>
<tr>
<td>5. Sample Weight and Dimension of Medical Cargo</td>
<td>41</td>
</tr>
<tr>
<td>6. Issues with Constructing a Low Cost ERT Robot</td>
<td>70</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Robot classification</td>
</tr>
<tr>
<td>2.</td>
<td>A medic robot</td>
</tr>
<tr>
<td>3.</td>
<td>The various mini-bots and a visualization of a swarm rescue</td>
</tr>
<tr>
<td>4.</td>
<td>A search and rescue robot</td>
</tr>
<tr>
<td>5.</td>
<td>The ReachBack robot diagnosing a victim</td>
</tr>
<tr>
<td>6.</td>
<td>A robot with a serpentine extension</td>
</tr>
<tr>
<td>7.</td>
<td>A modular train style robot</td>
</tr>
<tr>
<td>8.</td>
<td>A modular robot</td>
</tr>
<tr>
<td>9.</td>
<td>A raptor in flight</td>
</tr>
<tr>
<td>10.</td>
<td>A human-robot SR team</td>
</tr>
<tr>
<td>11.</td>
<td>Robot design</td>
</tr>
<tr>
<td>12.</td>
<td>Heat sensor beneath the head and heat avoidance path</td>
</tr>
<tr>
<td>13.</td>
<td>Internal cargo area</td>
</tr>
<tr>
<td>14.</td>
<td>Extendable chassis</td>
</tr>
<tr>
<td>15.</td>
<td>Layer architecture</td>
</tr>
<tr>
<td>16.</td>
<td>ERT decision modules</td>
</tr>
<tr>
<td>17.</td>
<td>State diagram for victim location and assessment</td>
</tr>
<tr>
<td>18.</td>
<td>Robot functionality</td>
</tr>
<tr>
<td>19.</td>
<td>ERT robot prototype</td>
</tr>
<tr>
<td>FIGURE</td>
<td>PAGE</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>20. IR emitter and the wire</td>
<td>67</td>
</tr>
<tr>
<td>21. A sample algorithm for supply distribution</td>
<td>73</td>
</tr>
<tr>
<td>22. A sample reasoning model to identify and locate supplies</td>
<td>74</td>
</tr>
<tr>
<td>23. A simulation of an ERT robot-victim interaction</td>
<td>75</td>
</tr>
<tr>
<td>24. Java Personal Agent model</td>
<td>76</td>
</tr>
</tbody>
</table>
ABSTRACT

A NON-INVASIVE ROBOTIC EMERGENCY RESPONSE TEAM

by

© Angie F. Shia 2006

Master of Arts in Computer Science

California State University, Chico

Spring 2006

Humans and dogs have traditionally accomplished Search and Rescue work. Recent years, with the terrorist attacks on Oklahoma and World Trade Center, robots have joined the team, particularly in the area of Urban Search and Rescue. Any site that is difficult for humans or dogs to access is suitable for small mobile robots. There is a very important piece missing in today’s Search and Rescue robots. Most of the current search and rescue robots focus on search – path planning, obstacle identification and avoidance, hardware design improvements, communication, multi-agent coordination, algorithms for finding victims and so on. However, very few, if any, focus on the rescue. This thesis proposes a type of autonomous and non-invasive SR robot called the Emergency Response Team robot. The main goal of the robot is to deliver emergency items such as bandage, water, food, and medication to the trapped victims. This thesis
presents the software and hardware designs required for such robots, in order for them to achieve the main goal.
CHAPTER I

INTRODUCTION

Humans and dogs have traditionally accomplished Search and Rescue (SR) work. In recent years, with the terrorist attacks in Oklahoma and on the World Trade Center, robots have joined the team, particularly in the area of Urban Search and Rescue (USR). Any site that is difficult for humans or dogs to access is suitable for small mobile robots. The Center for Robot-Assisted Search and Rescue (CRASAR), led by Dr. Robin Murphy of the University of Florida, is a dedicated federal/academic center for robotic-based SR [16]. During a speech at the 2002 American Association of Artificial Intelligence (AAAI) conference, Dr. Murphy made emotional and dismal assessments on the state of the SR robots. Based on her experience at the September 11 attack, she argues that the SR robots were completely disconnected from the real world. Her candid speech was not taken well by many who considered it harsh. Nevertheless, the main point is that it is vital that researchers and implementers of SR robots keep the robots connected with reality. After all, lives are at stake.

The Problem

There is a very important piece missing in today’s SR robots. Most of the SR robots are focused on search: path planning, obstacle identification and avoidance, hardware design improvements, communication, multi-agent coordination, algorithms for
finding victims, and so on. However, very few, if any, are focused on the rescue. The reason is unclear. Perhaps the medical rescue workers do not trust robots to administer first aid, or perhaps, performing rescue is difficult for an SR robot since it is usually only the small, agile ones that can get into a confined space. The first 48 hours is crucial to a victim’s survival. Rescue efforts may not reach all victims in that period. Furthermore, people that sustain injuries or a heart attack may not have 48 hours. This researcher found limited sources that addressed this need for rescue in current SR robots.

Relevant Work Done

SR robots touch on a wide range of topics. Usually a researcher focuses on a specific area of SR robot technologies. This thesis reviews only the topics that deal with different types of SR robots, their hardware and software, and their design features. Illustration is provided where appropriate.

The main purpose of this thesis is to suggest a design for a new type of SR robot called the Emergency Response Team (ERT) robot. The robot most similar to the one here suggested is the Reachback robot from CRASAR [15]. The thesis also reviews other robots and research from CRASAR [15], [16], [17], [18], from military, commercial and non-profit or government [8] research groups, and from academia [7], [9], [10], [11]. Some of the other relevant research that is not covered in the main chapters will be in the appendices.

Insight to Resolving the Problem

As noted by the Federal Emergency and Management Agency (FEMA), the first four to six hours after a disaster is critical to the survival of its victims. It is vital that
help is given to the victims during this time frame. Generally, a trapped victim dies from [1]: 1) difficulty in breathing, 2) dehydration, 3) bleeding, 4) medication, and 5) oversight.

**Difficulty in Breathing**

In a collapsed building, a fire may break out and smoke may develop. Gas pipes may burst from the collapse and create gas leaks. Other air problems include dust (concrete), toxic fumes, and any other agent that makes it difficult for a victim to breathe.

**Dehydration**

As in the World Trade Center (WTC) attack, the debris in the first few hours were removed by hand since heavy equipment may injure a trapped victim. However, hand removal of debris is very time consuming. It took months before the final clean-up of the area was accomplished. The WTC victims had no access to water. Even if the trapped victims have access to water, the water may be contaminated.

**Bleeding**

A victim may be struck by objects that fell from a shelf, or by concrete blocks from buildings that gave way. All of these can contribute to injuries. The victim can bleed to death (internally or externally) if first aid is not administered.

**Medication**

A victim may have specific medication needs, such as timely insulin injection or heart medication. Lack of medication causes secondary deaths.
Oversight

A victim may well be alive but unconscious due to any of the above factors. Rescue workers have no way to locate unconscious victims buried in piles of debris if the rescue dogs cannot find them in time or at all.

It is obvious that something must locate, access, and provide critical items to the victims. This something must be small, have basic “intelligence,” and can potentially communicate with the rescue workers. Dogs are still too big for this job. A small mobile robot is suitable for this task.

This thesis will only discuss the situation where the SR robot has lost contact with the base unit or an operator (human) and is completely alone with the victims.

Dr. Murphy and her students from CRASAR brought their experimental robots to the September 11, 2001 WTC site. In her debriefing article [60], Dr. Murphy discusses the various issues encountered at the site that were completely different from a simulated environment. The thesis attempts to address some of those issues in the proposed ERT robot and discusses how others address them.

Proposed Solution

This thesis proposes an autonomous and non-invasive ERT robot to assist victims as much as possible during the first four to six hours after a disaster. The proposed robot is not meant to administer first aid or treat injuries, even though such extensions may be included later on. The main goal of the robot is to deliver emergency items such as bandages, water, food, and medication to the trapped victims. FEMA has released a manual on confined space medicine (CSM). This thesis attempts to address some of the issues in CSM in relation to ERT robots.
The ERT robot hardware design includes the capability to move in the air and on the land. It has cargo space for carrying small, medical supplies. It has basic capabilities to protect itself (self-preservation). The chassis should be constructed with materials that protect the cargo against extreme environmental conditions. The chassis prevents the supplies from spoiling, melting, or being crushed. To locate victims, this thesis suggests using wireless techniques as an alternative method to computer vision. For demonstration purposes, the thesis will show how a victim could be located just using an infrared beacon.

The ERT robot software system design uses Artificial Intelligence (AI) techniques to make decisions and interact with victims. Techniques in Fuzzy Logic are introduced to resolve some of the issues in Natural Language Processing (NLP). A few Decision Support System (DSS) algorithms are suggested to evaluate victims’ conditions and dispatch supplies. However, if DSS is used in ERT robots, it has to be scaled down significantly since small mobile robots generally have limited memory and computation capabilities.

ERT robots are autonomous by default. Experiences and research from CRASAR have shown that disaster sites often create interferences with wireless communication. ERT robots must be untethered if a victim is located far beyond where any cable can reach. It is better for the robot to be unrecoverable if it means it has a chance to pass critical items to the victims. Hence, the material and construction costs for an ERT robot must be low.
Chapter 2 provides the necessary background information for SR robots. Topics include search and rescue, emergency response versus emergency medical team, software agents, mobile robotics, and medical decision software systems. Chapter 3 reviews some current search and rescue robots, and the people and groups involved in this area. It argues the need for an emergency response robot. Chapter 4 attempts to answer how to attain the goal of an ERT robot. This chapter discusses three issues affecting the goal: payload, hardware requirements, and software requirements. The payload issue is fundamental to the core of the thesis since the goal of the ERT robot is to carry supplies to the victims. Chapter 5 presents a proposed ERT robot design, reviews an ERT prototype using a non-visual method of locating victims, and discusses some of the issues involved in communicating with other robots. Chapter 6 recommends some areas of research in search and rescue robotics.
CHAPTER II

BACKGROUND INFORMATION

The purpose of this chapter is to provide the necessary background and overview to the topics discussed in this thesis. These topics include:

- What is search and rescue?
- What is the role of emergency response team versus emergency medical team?
- What is first-aid in confined space?
- What is a mobile robot and what role does it play in search and rescue?
- Who are the players in search and rescue robots?
- What are some medical decision systems that can be used in the SR robot?

Search and Rescue

Search and rescue (SR) is a broad term used to define a situation where agencies such as police and fire departments are involved during a crisis in which people are missing or trapped due to some form of disaster. This thesis focuses on search and rescue in urban settings. This is commonly known as urban search and rescue (USR). USR is different from regular SR efforts in the sense that the crisis environment is generally “man-made” (versus natural). A USR environment may typically contain collapsed buildings, toxic fumes, electric wires, cables, furniture, chemicals, iron bars, glass, and concrete, instead of rocks, snow, or water. Furthermore, there are also issues such as interference to wireless communications or electronic devices, toxic fumes, and potential...
explosions from burst gas pipes or chemical agents. These issues create problems to traditional search efforts and the equipment used.

Role of Emergency Response Team

Some companies or organizations have an Emergency Response Team (ERT). The ERT members’ role is to provide first level medical or rescue support to victims or evacuation efforts. Many companies or organizations solicit at least one volunteer from each floor or section to be trained as ERT members. These people are not members of the emergency medical team (EMT). ERT personnel are not medical professionals. They are generally a group of volunteers trained with basic medical, rescue, and evacuation knowledge and protocols. The rationale for appointing ERT personnel is that the first person at the scene is usually not someone from the fire or police department, hospital, or EMS, but is instead a bystander, relative, or colleague. A response within the first few minutes of many crises is crucial. For example, if an employee chokes on something or has a heart attack, even if the EMS show up within five minutes, the person can be in danger of dying or suffering permanent brain damage. If an ERT member is nearby, he or she can try to keep the victim’s situation under control before the medical teams arrive.

The ERT personnel are generally trained in the following [2]:

- Basic wounds and burns treatment
- Emergency breathing techniques
- Cardio-Pulmonary Resuscitation (CPR)
- Defibrillators usage
- Crisis (potential) identification
• Evacuation procedures

First Aid for Confined Space

FEMA has a handbook specifically on first aid in confined space (CSM) [1] for USR. Even though the handbook was written prior to the September 11 terrorist attack and focused mainly on earthquake victims, most of the ideas are applicable.

The handbook outlines the differences between typical emergency medical or pre-hospital care versus CSM. Some issues pertaining to CSM include personal performance problems, dangerous and limited access to victims, inability to remove victims, crush syndromes, hazardous material injuries, and airway dust impactions. The extrication site is usually chaotic. The victims and public are frightened. Frantic people overload the cellular phone satellites’ bandwidths. The site usually does not have sufficient medical backup personnel. Medical devices lose their communication capabilities from wireless interference.

Even though the first 24 to 48 hours is considered the most critical for survivors, the FEMA handbook also discusses the different types of victims: those that die within the first 4 to 24 hours versus those that can survive longer. For those that can survive beyond 24 hours, the handbook also outlines how to treat them. Timely extrication is vital but each victim’s environment is different and has to be handled accordingly.

The handbook also points out potential medical problems rescuers may experience such as fatigue, dehydration, dust, and potential exposure to toxic substances. In a Fire Engineering report [8], rescuers were actually poisoned by donated food because the food was sitting under the hot sun for a long time.
Site analysis must be done quickly. Collapsed buildings are also categorized into different types – pancake, lean-to-floor, V-shape, and cantilever. Each type requires a specific stabilization method before rescue can begin.

Atmosphere monitoring is also crucial as gas pipes, toxic chemicals, dust and many other life-threatening fumes may be present or potentially exposed.

The handbook identifies five different roles or functions in a USR first aid unit, as quoted from the handbook [1, p. 28]:

Provider – The “point person” having “hands-on” contact with victim:
- Performs evaluation/treatment
- Thinks and acts “out loud”
- The provider at the patient’s side may vary depending on multiple parameters:
  - Physical characteristics, size of space/personnel. Strength need to move patient etc.
  - Technical needs – physician procedure/assessment vs. medics immobilization and extrication skills
  - Personal – fatigue, heat sensitivity, etc.

Anticipator:
- Person listening to the evaluation/treatment
- Communicates anticipated needs of the Providers.
- Prevents “dead time” with the providers awaiting equipment/supplies
- Ask “what’s needed” frequently
- Assure equip/supplies are delivers and prepped.

Medical Control MD:
- Physician providing orders if the providers are Medical Team Specialists
- May also be the Anticipator

Recorder/Safety Officer:
- It is vitally important that a member be assigned this function.
- Records “times” assessments, intervention, repeat vital signs and patient re-assessments.
- Assures atmospheric monitoring reading if indicated; assures recheck of “roof” and other safety precautions.
- Monitors providers length of time are “in the hose” and so need for rotation, etc.
- Track amount of O2 in tanks, etc.
- Assume # of providers in and out of space are equal, records type and number of feet of rope into space, etc.

Equipment/Supply Officer:
• Person obtaining needed equip/supplies.
• Prepares (pre-assembles) them for immediate use by the providers.
• Monitors expenditure of cache supplies.

These functions are important to keep in mind when designing an ERT robot. This thesis explores the role of SR robots within these functions.

Overview of FEMA’s Urban Search and Rescue Agency and The Center for Robot-assisted Search and Rescue

CRASAR is a (Type II) dedicated federal and academic center for robotic-based search and rescue. CRASAR is part of the University of South Florida. Dr. Robin Murphy is currently the director of CRASAR. CRASAR was originally established with a connection to the National Institute for Urban Search and Rescue (NIUSR) [65]. They work closely with various government agencies, particularly the Urban Search and Rescue (USR) from FEMA. Even though there are many universities [7], [9], [11], [53], [66], [67], [68] conducting research on search and rescue robots, CRASAR is different because it is also a response center. This means that they may also deploy people and robots to a disaster site. Today, CRASAR performs research, evaluation, deployment, and training on all robotic related search and rescue operations.

Dr. Murphy and her students brought their then experimental SR robots to the September 11, 2001 WTC site. In her debriefing article [60], she discusses the various issues met at the site, which were completely different from a simulated environment. Her review of the WTC experience identified many oversights of the then state-of-the-art SR robot designs. Her review may have changed the paradigm of SR robot design altogether.
Introduction to Mobile Robots

What is a Robot?

This is a difficult and contentious question in the sense that the word *robot* means different things to different people. The answer not only differs from one person to another, but also from one field to another. Wikipedia defines a robot as [57, p. 1]:

In practical usage, a robot is a mechanical device which performs automated physical tasks, either according to direct human supervision, a pre-defined program, or a set of general guidelines using artificial intelligence techniques. Robots are typically used to do the tasks that are too dirty, dangerous, difficult, repetitive or dull for humans. This usually takes the form of industrial robots used in manufacturing lines. Other applications include toxic waste cleanup, underwater and space exploration, mining, search and rescue, and mine finding. Recently however, robots are finding their way into the consumer market with uses in entertainment, vacuum cleaning, and lawn mowing.

Merriam-Webster online dictionary defines it as [58, p. 1]:

1 a : a machine that looks like a human being and performs various complex acts (as walking or talking) of a human being; also : a similar but fictional machine whose lack of capacity for human emotions is often emphasized b : an efficient insensitive person who functions automatically
2 : a device that automatically performs complicated often repetitive tasks
3 : a mechanism guided by automatic controls

Whichever definition one uses, the general idea is that a robot is a mechanical object control by something or someone and performs some function. It can be autonomous (think, act, or move about on its own).

What is a Mobile Robot?

A mobile robot is a specific type of robot that can move about on its own. It may be controlled remotely or by an on-board program. A mobile robot generally has three main characteristics: a) it has effectors, things to allow it to grab objects, such as a arm and hand; b) actuators, things that allow it to move, such as wheels; and c) sensors,
things that allow it to gather data, such as a camera, thermometer, or pressure gauge [55]. Mobile robots can be:

- Operator-controlled—A robot is generally fully controlled by a human operator.
- Tele-operated—A robot may be partially controlled or supervised by humans.
- Autonomous—A robot thinks, acts, and runs around on its own through an onboard program.

Some of the “famous” mobile robots are the Mars Polar Lander and Pathfinder space mission robots from NASA/JPL [69], and the ROOMBA vacuum cleaner from the iRobot company [82].

SR robots are a type of mobile robot. The equipment and payload they carry and their decision-making systems are medical in nature and catered to saving lives.

The research developed and the technologies used in all mobile robots are generally interchangeable. This is because they focus on improving the common hardware and software needs. The common hardware needs are better chassis, effectors, and actuators. The common software needs include better path planning, obstacle avoidance, decision-making, and image processing.

**Classification of Mobile Robots**

Table 1 shows the different classification of robots [52] based on weight and size. A robot may fall between categories.

<table>
<thead>
<tr>
<th>Cubic Inches</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

**TABLE 1**

CLASSIFICATION OF ROBOTS
<table>
<thead>
<tr>
<th>Weight Range</th>
<th>Bot Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 lbs &lt; x &lt; 50 lbs</td>
<td>mini-bot</td>
</tr>
<tr>
<td>50 lbs &lt; x &lt; 100 lbs</td>
<td>macro-bots</td>
</tr>
<tr>
<td>100 lbs &lt; x &lt; 300 lbs</td>
<td>maxi-bots</td>
</tr>
</tbody>
</table>

Generally, the two extreme sizes (micro and maxi) are used in search and rescue. Micro robots make good confined space bots, while maxi robots are typically used as chauffeurs to port smaller robots around.

**Autonomous Agents and Robots**

The decision to implement robotics software as agent systems versus regular programs has a major impact on robot behaviors. Loosely defined, a software program is an instruction set for computer hardware that computes some data. Whereas a software agent is a specialized software program that has the ability to *reason*. Software agents generally embody some combination of AI techniques or principles such as Natural Language Processing [59], Automatic Speech Recognition (ASR) [59], and decision-making [33]. The object that the agent represents could be anything. One interesting feature of software agents is that they are autonomous [44], meaning they can reason on their own without human intervention.

These agent programs are the “robot brains” when they are used in robots. These programs may control or execute other non-agent based programs. There has been much research in the software agent area, particularly with multi-agents or distributed agents [45], [46], [47], and [48].
The Need for Autonomy

In Chapter 1, a concern was pointed out that pertains to situations when robots lose contact with their base either through signal interference or equipment failure. From a survey conducted by CRASAR, there is a distrust of robot autonomy by the rescue and medical personnel. Hence, CRASAR has redirected their research focus to create operator-controlled robot systems [15]. This thesis raises this issue and suggests that autonomy is vital to search and rescue robots and should still be researched. Naturally, when it comes to saving lives, it is wise to err on the side of caution. However, overly cautious behavior may also prevent one from saving lives. A compromise may be possible in that limited autonomy with minimal side effects can still be pursued.

Medical Diagnostic System and Rescue Robots

Many medical diagnostic systems used in hospitals or medical facilities use AI techniques. Traditionally, medical diagnostic systems are expert- or case-based reasoning (CBR) [5], [34], [35] systems. In recent years, researchers incorporated probabilistic models such as the Bayesian theory into CBR systems [49], [50], [51] to handle complex and uncertain domains and real-time environments. The model introduced by Vicari et al. specifically deals with medical knowledge in a dynamic, multi-agent environment [49]. In the future, these types of systems could be part of the medic robots such as the ERT robots proposed in this thesis. This would add tremendous value to rescue operations.
In recent years, research on USR robots has proliferated in various groups from academia, private companies, and government (military). This chapter provides an overview of some of these robots and their features.

This chapter is split into four sections. First, it reviews the general classification and requirements for SR robots. Second, it reviews some current USR robots. Third, it discusses the issues involved with these robot designs and the problems in the current search and rescue paradigm. Finally, it presents the need for a special type of robot called an ERT robot.

Classification and Requirements for Search and Rescue Robots

CRASAR provides detailed requirements for all SR robots [16]. This thesis uses the requirements as a reference. The following is a general outline of SR robots requirements from CRASAR:

- It can fit into a backpack.
- It can be checked in as luggage on airlines.
- It can be operated in sunlight and in darkness.
- It can be easily decontaminated.
- It can provide APIs for software development.
Furthermore, CRASAR indicates every robot should also have the following features, sensors, and payload:

- Color video camera with zooming feature
- Two-way audio
- LED lighting
- A tether or a latch for rope
- Waterproof chassis
- Invertible or self-righting chassis
- Temperature sensor
- A MTBF (Mean Time Before Failure) 96 hours system.
- An ISO 9001 compliant system
- Easy-to-spot color chassis
- Wireless communication

Figure 1 from Micire’s thesis [37] shows how a search and rescue robot is classified. The classification is based on the environmental characteristics, the robot dimension, and the task requirements. For example, the robot’s size, communication method, and battery life determine the size of void and its payload. The moisture level at the site affects the robot’s traction.

Urban Search and Rescue Robots Overview

For simplicity, this section splits the SR robots into three categories based on their hardware design. The thesis will hereafter refer to USR robots using these categories below.

- **Cart-and-wheel** – These are the “classic” types of robots. They have wheels as effectors and a body (chassis) attached to the wheels.
• **Biology-inspired (serpentine and crawlers)** – These types of robots generally resemble some kind of animal or legged insect. The serpentine robots resemble snakes or snake-like creatures. The crawler robots resemble legged-insects such as ants or spiders.

• **Aeronautical** – These types of robots can fly or ascend from the ground.

**Cart-and-Wheel Robots**

There are seven cart-and-wheel style robots: the medic robots from Applied Perception [23], the mini-bots from Sandia Laboratory [22], the TALON robots from Foster-Miller [68], the Packbot robots from iRobot [67], the SR robot from Carnegie Mellon University [86], the Medical Reachback robot from CRASAR [15], and the Security robot from Mostitech [71].

**Medic robot from Applied Perception.** Applied Perception originated from Carnegie Mellon University. The medic robots from Applied Perception resemble lawn-mowers. They are used as medic robots mainly in a battlefield. The robots drag wounded soldiers to a safer area. Another larger, car-sized robot is designed to transport the soldier elsewhere [23]. Figure 2 shows the robot listed on their website.

**Swarm robot from Sandia National Laboratory.** Sandia National Laboratory’s approach to search and rescue mission is to use numerous tiny robots. They create small, dime-sized robots [22] called mini-bots. Figure 3 shows the different types of mini-bots.

The mini-bots runs on a type of “swarming” algorithm called the “Distributed Optimization” algorithm. The algorithm provides group intelligence for a swarm of mini-bots. It can rapidly pinpoint a source of contagion and is suitable for Hazmat missions.
Anyone that carries a minicomputer, a global positioning device (GPS), or simple radio equipments can use the mini-bots. The robots that use swarming algorithms have been demonstrated to find avalanche victims four times faster than any other published search scheme.

**TALON robots from Foster-Miller.** The TALON robots are human-portable robots [68]. They are widely used in military operations, rescue missions, or bomb defusing missions. Some models have on-board ammunition. The TALON robots have human override keys that shut down runaway robots. Some models are also large enough for a human driver.

**PackBot robots from iRobot.** Packbot robots are direct competitors with TALON robots [67]. Their uses and functions are similar. The Packbot EOD is used in
search and rescue operations. Packbot EOD is more lightweight than the TALON. It is teleoperated. The body has a spooler that prevents the cable from tangling.

Search and rescue robot from Carnegie Mellon. Carnegie Mellon University (CMU) showed an interesting design during the 2003 RoboRescue event [86]. Figure 4 shows the CMU robot. The robot has two bicycle-size wheels. The large wheels enable the robot to climb stairs, a task that a standard tracked-wheeled robot cannot perform. The CMU robot wheels also have a tight turning radius. The robot has a tail to help keep it balanced. The robot sensors and system board are located between the wheels.

**Medical Reachback robot from CRASAR.** The Reachback robot [15] design is a standard cart-and-wheel robot. However, the robot is also a medic robot. It carries a sophisticated triage sensor developed by CRASAR. It also carries intravenous (IV) and oxygen tubings to trapped victims. This thesis revisits this robot later as it is very similar in idea with the proposed ERT robot. Figure 5 shows the robot in action during a mock disaster demonstration.

**Security robot from Mostitech.** The final robot using the cart-and-wheel design is a Mostitech robot [71]. The robot is technically not an SR robot but a home security robot. The robot can move about on its own. It has the ability to identify an “emergency.” During an emergency, it takes pictures using its on-board camera and then sends them to the owner’s cellular phone.
The Mostitech robot can be modified into a simple SR robot for the home or it can be modified to communicate with other SR robots. The Mostitech robot can provide valuable information such as where a victim was last seen in the house. It may also be able to access a victim if rescue robots cannot get into the house.

The weight of a Mostitech robot is about 25 pounds and its height is around 20 inches. The robot costs surprisingly little for what it can do. Its retail price is only $850 dollars. Its nearest competitor costs over $10,000. Later, Mostitech intends to add facial recognition technologies to its robot.

**Biology-inspired Robots**

Many robots are inspired by biology; that is, they are built to look like an animal or insect and have certain characteristic of that animal or insect. For example, serpentine robots look and act like animals that move on their belly, such as snakes and worms, while multipods robots look or act like legged insects such as ants. Most biology-inspired
robots cannot carry much payload, since they are designed to maneuver in tight spaces. Hence, they make good reconnaissance robots but not good rescue robots.

**Serpentine robots from Robotics Institute.** Choset from Carnegie Mellon [66] proposed a serpentine style robot for use in search and rescue. He argues that serpentine mechanisms have more degrees of freedom than the cart-and-wheel style from the previous section. This is because many disaster sites are narrow and constrictive. Traditional robots have difficulty accessing such sites. Figure 6 shows one of their serpentine robots.

**USAR robot from US Naval Academy.** Hudock et al. from the United States Naval Academy (USNA) created a snake like robot for search and rescue [4]. They examined a new way to design a serpentine style search and rescue robot (Figure 7) by refining Choset’s snake idea [66] and changing the robot from a snake to a train design. For software, USNA uses genetic algorithms and Open Dynamics Engine (ODE).

**Modular robot from Xerox.** The Palo Alto Research Center (PARC) team took the serpentine style to a completely new level. Their modular robots (Figure 8) are connected by individual modules [11]. These modules reconfigure themselves based on AI techniques, or they can be human-configured. The interesting part of the modular robot is that each module can be individually programmed. This not only increases the robot’s versatility dramatically but also increases its complexity. The software has to include code that prevents the modules from physically colliding with one another. According to the paper, a point-to-point self-collision check for a modular robot with n modules requires a time complexity of \(n^2\). The PARC team argues that in order for
modular robots to be successful, issues such as programming cost and complexity must be resolved [11]. The team is working on a 200-module polybot.

![Image](image.jpg)


**Ant robots.** Massachusetts Institute of Technology’s (MIT) Artificial Intelligence Lab built a robotic ant colony [83] to study colony behavior. The cubic-sized robots mimic the real ants in finding food and task cooperation. For the robots’ system, they developed the Antware [83] using the Subsumption Architecture programming style developed by Brooks [85]. Each robot’s system reflects a certain behavior. These behaviors are tied to the robot’s action and are organized in hierarchically format. A more “important” behavior can override a lesser behavior and its pending action.
Wagner from the Israel Institute of Technology (Technion) has researched how ants and other insects use pheromones to communicate and coordinate various tasks such as finding food [84]. He has developed multi-agent systems using this principle. The ant robots running on this system were able to complete two out of three tasks in polynomial time.

**Aeronautical Robots**

Aeronautical robots are robots that can fly. Most resemble small helicopters. Aeronautical robots are commonly known as Vertical Take-Off and Landing (VTOL) robots or Unmanned Aerial Vehicle (UAV). VTOL robots and UAVs are nimble and can maneuver around tight spaces. Their obvious strength is that they can fly over obstacles and ravines. This is something the cart-and-wheel and serpentine robots have extreme difficulty in overcoming.
The downside of the aeronautical robot design is that its robots have to lift its payload as well as its own weight. This is a problem the cart-and-wheel and serpentine robots do not have. If a VTOL robot does not have a gas-powered engine, it will not be able to carry much payload [70]. Most aeronautical robots are used in reconnaissance or search missions.

**Raptors from CRASAR.** Garcia et al. [19] designed an effective vision system for aeronautical robots. They built two miniature helicopters called the Raptor 70 and Raptor 90. The maximum payload for the robot is about 8.5 pounds. The robots have an on-board camera and GPS system. Figure 9 shows a Raptor from Garcia’s paper.

Flying robot from Waterloo Aerial Robotics Group. Waterloo Aerial Robotics Group (WARG) built various types of aeronautical robots [87]. Each robot has an intelligence system that is part of a distributed, remote vehicle network. Each vehicle represents a node on the network and fulfills a specific function. The nodes are arranged hierarchically. WARG created a four-level remote vehicle network that consists of a base station, a transport vehicle, a delivery vehicle, and a payload vehicle.

Issues with the Current Designs and Search and Rescue Paradigm

This section presents an overview of various types of search and rescue robots. Each of them has their strengths and weaknesses. With the exception of the ReachBack robot from CRASAR, none of the other SR robot designs focused on basic medical rescue. In Micire’s thesis [52], there were many SR robots from various groups
helping with the search and rescue efforts. It would seem prudent to organize these robots into specific functions to make the effort more efficient. Figure 10 offers such an organization.

![Diagram of human-robot SR team]

An SR robot team must also mimic an “organic” rescue team. In an organic team, each member, human or dog, plays a specific role. An SR robot team would be comprised of robots from various designs. Each robot is assigned a specific role that reflects their strength. For example, a VTOL robot would be a scout, while the CRASAR reachback robot could be a medic.

A human SR leader can assign a generic assignment for robots. The robots can then split this assignment among themselves. The robots with similar hardware designs can take on a specific function of the assignment. They could further split a function into many sub-functions and distribute the workload among themselves.

Together, all of the SR robots collaborate and contribute their efforts to the main goal (assignment). This suggested SR team paradigm is similar to the WARG robot network. The idea of distributed workload and function is essentially the concept behind
multi-agent systems [45], [46], [47], [48] and object-oriented design [72] in programming.

The Need for a Special Rescue Robot

Under the human roles or functions listed in Chapter 2 and shown in Figure 10, there is a function called Supply Officer. However, there is no specification on who is going to deliver the supplies. Figure 10 suggests a new robot role called a Supply Deliverer. This role can be fulfilled by a small robot. Currently, with the exception of the Reachback robot, none of the other robots reviewed above performs rescue.

It is possible to design a simple, low risk, rescue robot that only focuses on delivering medical supplies. This section revisits the FEMA handbook on what victims need and suggests how a special type of robot can indirectly address those needs. This thesis hereafter refers to this robot as an ERT robot.

What Do Victims Need?

Chapter 1 argued that the first 4-10 hours for a victim is crucial based on FEMA’s analysis [1]. It takes an average SR robot somewhere between 10-20 hours to find victims [15]. There has to be something that can be done for these victims; even if it is temporary or minor, like passing comfort food or a toy for a child. Since there is a high possibility that a search and rescue robot may be the first (or last) to access a victim, it would be helpful that the robot carry something to help victims.

Chapter 1 mentioned some of the issues a victim may face:

• Difficulty In breathing

• Dehydration
• Bleeding
• Medication
• Oversight

Table 2 shows the issues on the left column and some remedies on the right.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Remedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Difficulty breathing/toxic air</td>
<td>Oxygen or clean air</td>
</tr>
<tr>
<td>Dehydration</td>
<td>External water (internal may be contaminated)</td>
</tr>
<tr>
<td>Bleeding</td>
<td>Bandage or bleeding stopper</td>
</tr>
<tr>
<td>Medication</td>
<td>Personal or general</td>
</tr>
<tr>
<td>Oversight</td>
<td>ID Tag</td>
</tr>
</tbody>
</table>

What Can the Robot Carry?

What can an SR robot carry and how would it accomplish that? An average SR robot is about the size of a shoebox [16], [17]. The content of its payload is dependent on weight and size. Some content may even be dangerous in an unstable environment.

Below are various methods to resolve the issues mentioned above. Note that this section assumes victims are conscious and can help themselves. Another section discusses unconscious victims.

**Difficulty breathing/toxic air.** In a disaster site, the air may be filled with smoke, gas, or toxic fumes that deprive victims of oxygen. However, a robot cannot carry
in an oxygen tank to victims. Aside from weight and size constraints, oxygen tanks can explode under pressure.

The ReachBack robot from CRASAR tried to get around this problem by dragging IV and oxygen tubes to victims [15]. There are two obvious problems with this method: first, the tubing may not be long enough to reach the victim; second, the tubing may melt. The site may have a fire or an explosion that makes the area very hot. Another potential problem is that lengthy objects such as wire, cables, and tubing tangle easily. If the tangle is inside the trapped area, the rescue workers may not know that victims are not getting the oxygen or water they need.

Instead of dragging tubing like the Reachback robot, an ERT robot can carry gas masks. Unlike the oxygen tubing, the masks are not exposed to the environment but are inside the robot chassis. To prevent the masks from melting, the chassis of the ERT robots must be fire-retardant and able to keep the internal temperature low. Chapter 4 discusses these (robot) hardware design considerations.

There are many foldable gas masks products available today. One such type of specialty gas mask is called the Potomac gas mask. The Potomac mask is lightweight and foldable. It can filter most toxic fumes for at least 15 minutes [32]. A victim could use more than one mask to extend the filtering time.

**Dehydration.** As mentioned, a robot should not carry an IV tube. Instead, a more urgent requirement of water must be carried in. An average person can survive without water for 3 or 4 days [1], [2]. If the victims are located, they can be given an endless amount of water for each robot “run.” That is, once a path has been identified (and possibly tagged with some form of path beacon) between the victim and the rescue
area, an ERT robot can run back and forth with the water. However, if the robot cannot run back and forth, they should at least carry a small water pack. An 8 oz. water pack can easily fit into a small robot chassis.

**Bleeding.** If a victim is bleeding, the simplest way to slow it down is to use a bandage rubber tie [2] or tourniquet [73]. These items are light and can easily fit into a robot chassis. As mentioned in Chapter 1, ERT robots do not administer first aid but to deliver supplies. The tourniquet and bandage is for victims that are conscious and can help themselves.

**Medication.** The benefit of bringing in medication for victims is innumerable, especially for victims with heart conditions or diabetes. However, there is an issue to consider when bringing in medication. Medication that a general population can use, such as painkillers, would not pose a significant health risk. However, bringing in prescription medication is riskier.

There are two problems with giving victims prescription medication. First, a medical diagnostic system is required to prescribe medication. Second, a victim identification system is required to avoid giving the wrong medication to the wrong victim.

For the first problem, if the victim did not have a prescription medication before and started to exhibit symptoms that require it, the robot has to diagnose the victim. As mentioned in the first chapter, this thesis assumes that the ERT robot is not able to communicate with a human operator. This means that the victim diagnostics would be performed solely by the robot system. Even though medical diagnostic systems are common in hospitals, this is not common in SR robots [15]. If the victim is unconscious, it can increase the risk of an incorrect diagnosis. Since the survey conducted by CRASAR
[15] shows that the medical and rescue personnel do not even like the idea of SR robots making search and rescues autonomously, one can conclude that the idea of adding on a non-human controlled medical diagnosis system for the victims will not be well accepted.

For the second problem, it is vital that the robot correctly identify a victim for a medication. This requires an accurate victim identification system such as a facial recognition program. However, disaster sites may have insufficient lighting. This can affect the accuracy of facial recognition systems [74]. This capability requires more research.

Until the issues involving prescription medications are resolved, an ERT robot should only carry-over-the-counter medication. If the medications are in pill form, the robot can easily carry a hundred of them.

**Unconsciousness.** An ERT robot should not do anything to an unconscious victim. Chapter 4 discusses the ERT protocols [2] such as checking a victim’s reflex through light bumping of the victim, broadcasting loud sounds, and other assessments. If the robot has tried all the procedures and a victim is still not responding but is believed to be alive (for instance, their vital signs are detected), the robot can leave some form of identification tag for the excavation crew if another SR robot has not done so (radio frequency identification tags can be used for this purpose). The ERT robot should then proceed to the next victim.

**Looking Ahead**

Chapter 4 attempts to answer these three questions regarding the ERT robot:

- How ERT robots locate victims—the chapter suggests using non-vision based systems and demonstrates a victim localization method using infrared signals.
• How ERT robots assess a victim’s condition or approach a victim—the robots will follow standard victim assessment protocols. The ERT robot systems should include programs that exhibit certain behaviors for the robot.

• How ERT robots decide what to give to victims—there are two issues to consider—first, victims may demand items they do not need or take all of the supplies for themselves; second, victims may not know what they want or are too confused to make decisions. This requires an on-board decision support system (DSS).
CHAPTER IV

EMERGENCY RESPONSE TEAM ROBOT

REQUIREMENTS AND SPECIFICATIONS

This chapter discusses the requirements and specifications for an ERT robot: which contents should be in the robot’s payload; the main hardware requirements for an SR robot and a proposal for a hybrid design to meet those requirements; and some of the issues involved when designing an ERT robot system.

Payload Requirements and Specifications

There are various issues involved in payload calculation. When calculating payload, the weight of a robot must also be included, because the motors and batteries will expend a portion of their energy on moving the robot. The choice of chassis, battery, and on-board components will affect the weight of the robot. Furthermore, the physical size of an ERT robot will be small and have limited cargo space. Since there are weight and size constraints, it is important to carry only necessary items that are not bulky.

Feasibility versus Criticality

Critical items for a small mobile robot. A basic mobile robot requires items such as batteries, actuators, effectors, motors, wires, and a system board [29]. The actuators allow it to move. The effectors enable it to grab things. The motors control the effectors and actuators. A system board runs its programs. Finally, a battery supplies the electricity to all of the electrical components.
Mobile robots with different functions require different sensors. An SR robot would need extra features such as path planning or decision making programs and special sensors to look for victims. It may also include sensors and programs to help protect itself from hazardous environments.

In general, an SR robot needs headlights, heat sensors, a motion sensor, and an intelligent program. The Reachback robot from CRASAR [15] has a triage sensor. This may be the *de facto* standard for future medic robots.

Many SR robots are also equipped with cameras (computer vision). Computer vision is excluded in this thesis as a minimum requirement because the victim identification using computer vision has not reached a reliable level [30]. For example, how does a robot determine, from vision, if a victim is dead or unconscious? Furthermore, who is doing the viewing? If the camera were meant for the human operator, it would make sense; however, if the robot were autonomous and does not have a human operator, the triage sensors [15] used in the ReachBack robot would make more sense.

**Feasibility.** Table 3 shows the minimum space and weights required to construct a mobile robot. Column one lists the average robotic components that one can find on the web [31] or electronic stores. The second and third columns show the average weights and dimensions of these materials. For simplicity, the materials are assumed to be stacked vertically, or, placed on their side next to one another, horizontally. In real life situations, more space efficient ways of organizing these materials are possible. The dimensions shown in the second column of the last row is computed using the largest length and width of all the dimensions and their total heights. The third column is the sum of all the weights.
TABLE 3
MINIMUM SPACE AND WEIGHT REQUIREMENTS
FOR A BASIC MOBILE ROBOT

<table>
<thead>
<tr>
<th>Item</th>
<th>Dimension</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 NiMh battery (3 hours)</td>
<td>3” x 4” x 1”</td>
<td>16 oz</td>
</tr>
<tr>
<td>1 system board</td>
<td>3” x 4” x 1”</td>
<td>6 oz</td>
</tr>
<tr>
<td>2 metal wheels</td>
<td>2” radius</td>
<td>2 oz each</td>
</tr>
<tr>
<td>1 actuator (pincher arm)</td>
<td>5” length</td>
<td>6 oz</td>
</tr>
<tr>
<td>1 sturdy metal chassis</td>
<td>5” x 3” x 3”</td>
<td>8 oz</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>5” x 4” x 5”</strong></td>
<td><strong>38 oz</strong></td>
</tr>
</tbody>
</table>

Table 4 shows similar data for an SR robot as described previously. The asterisks highlight the extra items required for an SR robot. Note that an extra EEPROM chip is added, as an onboard AI program will probably be computationally intensive. A regular system board like the Board of Education (BOE) used by the Parallax BoeBot [29] will not be sufficient.

So far, the total approximate weight for an SR robot is less than three pounds. Its space requirement is less than the size of a Christmas card box. Table 4 only shows the minimum requirements. There are other important items such as toxic fume monitors, gas monitors, and triage sensors.

TABLE 4
MINIMUM SPACE AND WEIGHT REQUIREMENTS FOR A BASIC SEARCH AND RESCUE ROBOT
Chapter 3 presented various types of SR robots. They were either angular like a shoebox (cart-and-wheel style), or lengthy like a snake, or agile like a helicopter. If a shoebox is assumed to have a dimension of 12 x 8 x 6 cubic inches, the critical components of a small cart-and-wheel robot can occupy up to half of the space. That leaves only the other half for cargo space. The number of basic medical items that the cargo area can hold depends on how the area is designed. The space and weights of these items are listed in Table 5. The table uses the same calculation methods as in Tables 3 and 4.

**TABLE 5**

SAMPLE WEIGHT AND DIMENSION OF MEDICAL CARGO

<table>
<thead>
<tr>
<th>Item</th>
<th>Dimension (l x b x h)</th>
<th>Approximate weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 foldable gas mask</td>
<td>6” x 8” x 2” (boxed)</td>
<td>8 oz</td>
</tr>
<tr>
<td>Item</td>
<td>Dimension (l x b x h)</td>
<td>Approximate weight</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>1 water pack</td>
<td>3” x 4” x 2”</td>
<td>8 oz</td>
</tr>
<tr>
<td>1 duct tape</td>
<td>1” x 2” x 0.10” (wallet size)</td>
<td>2 oz</td>
</tr>
<tr>
<td>1 bandage/gauze</td>
<td>1” x 2” x 1” roll</td>
<td>2 oz</td>
</tr>
<tr>
<td>1 pill box</td>
<td>2” x 2” x 1” (about 100 pills)</td>
<td>4 oz</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>6” x 8” x 6.1”</strong> (maximum of Ls, Bs and sum of Hs)</td>
<td><strong>24 oz</strong></td>
</tr>
</tbody>
</table>

The problem with cart-and-wheel is that it is still large for tight spots. The slender design of the serpentine robot does not leave much room for cargo. The Raptor robots from Chapter 3 have limited space but can carry up to 8.5 pounds of cargo. However, the space constraint in the Raptors (helicopter style) creates a problem, as some of the critical items are bulky.

**Other Issues Affecting Payload**

Many of the extrication locations are hazardous. Some areas may have fires or a recent fire that heats up the concrete blocks from fallen buildings. These heated blocks can melt the chassis of the serpentine robots or the wheels of the cart-and-wheel robots. There may be pools of water or flooding at some areas. If there are live electrical wires dangling near them, an SR robot can be short-circuited. Dust can also interfere with an SR robot’s system board and other delicate computer sensors.

The content of the payload must be able to withstand such extremities. Chapter 3 argued against bringing in an oxygen tank into a site. The proposed alternative (gas mask) is much safer. However, it can melt under high temperature. Thus, for payloads
that do not pose a danger but will break, spoil, melt, or just not function under extreme environments, the robot chassis has to protect the payload from above problems.

Hardware Design Requirements and Proposal

As argued in Chapter 3, an SR robot team should be comprised of different types of robots, each with a different function. The proposed ERT robot is primarily a “medical cargo delivery” robot.

Understanding Hardware Design Requirements

Here five basic hardware considerations for a robot: tether-control, sensors, communication, system board, and chassis.

Tether versus non-tether control. In many SR robots today [16], [18], tethering (to a cable or line) is common for a few reasons. First, it prevents wireless communication interference. Secondly, cables and T1 lines have high bandwidths [81]. This is important when voice control, camera, and image processing are part of the data sent back by the robot. Thirdly, it allows direct operator control. A tether also prevents the robot from getting lost. An operator can always retract the cables to retrieve the robot.

However, tethering a robot has some problems. For example, it has the same problem as dragging in an IV or oxygen tube. The cable can snag on something, can tangle, melt, and break. Furthermore, the robot cable may not be long enough. The debris at disaster sites can pile up to 30 feet high, which would require SR robots to maneuver through long, windy paths before they reach disaster victims. Non-tethered controlled robots, whether they are scouts or mission robots, are not widely accepted by rescue teams at this point [15].
The ERT robot design is therefore not going to deviate from CRASAR’s tether requirement for control of a robot. However, in the ERT robot design, tether control is not mandatory. It is an “override” feature. If an ERT robot is tethered, the robot system immediately relinquishes control of the robot to the human operator. This in turn shuts down or stops the on-board AI program or decision support system (DSS) and routes the data to the operator.

The ERT robot is expected to be an autonomous robot. As noted in Chapter 2, fully autonomous SR robots are currently not trusted or accepted. Nevertheless, research should continue in this area. There may be situations where the robot loses contact with the operator due to either a broken or failed cable link, or some other incident where the robot is considered Missing In Action (MIA). The robot must have some kind of on-board program to tell it what to do (default execution program). It can either immediately shut itself off in case of endangering others or go to some pre-defined coordinates and so forth. As the Mars exploration by NASA has shown [51], two $150 million probes were lost and mission control could not contact them. Fortunately, both probes had on board default execution programs.

An ERT robot executes a default medical protocol program, searches for victims, and then passes supplies to them. One may be concerned with runaway robots that pose danger to the public. Chapter 3 emphasized that the purpose of ERT robots is to leave supplies, and not administer first aid. Therefore, the robot’s function is to identify a human target and leave the medical supplies. Thus, if for some reason, an ERT robot were to become a “runaway robot,” the worst-case scenario would only be the robot misidentifying an inanimate object as “human” and loses its medical supplies. The robot
would then either shut itself down, return to base, or go to a rendezvous point. These types of risk mitigation strategies must be included in the software design for an ERT robot.

**Sensors required.** The following are the basic sensors required for the ERT robot and their respective functions.

- Infrared—detects heat. Used mainly for self-preservation and identifying potential victims.
- Ultra-sound (sonic) —detects sound.
- Pulse sensor—detects heartbeats (live victim).
- Voice recorder—identifies a victim or enables a victim to leave messages.
- Camera—helps in path planning, obstacle avoidance.
- Whisker—helps in path planning, obstacle avoidance, and space estimate for traveling.
- Communication device (wireless or tethered cable) —enables robot to interact with humans.

**Communication.** As mentioned in above, the robot must allow tether-over-rides. However, in a situation where the cable is tangled or disconnected from the robot, the robot must have wireless communication capability to communicate with the human operator. However, communication may not be possible if disaster sites have wireless communication interference.

Other protocols may include inter-robot communications. Chapter 5 discusses some of the problems with such communications.
**System board.** The on-board micro-controller must be able to handle large programs, as ERT robots may require programs for path planning, obstacle avoidance, sensor data fusion, and a medical system. The actual on-board programs will vary from one implementer to another.

There should be at least a 64M RAM micro-controller. A simple AI program would suffice for an ERT medical system. The other on-board programs in an ERT robot should be kept to a minimum.

Complex AI programs may require a small motherboard or multiple micro-controllers. The next section discusses software issues in micro-controllers. It also compares the differences between using a micro-controller and a motherboard in a mobile robot.

**Robot chassis.** The robot chassis must be lightweight and heat proof. The material used should have a low density but a high melting point. Composite materials are probably more suitable than metal. Some of the robots used in the popular Robot Wars [75] contest even have fire retardant (as those used in firefighters’ coats) coating on their chassis.

**ERT Robot Design**

Figure 11 is an artistic rendition of the proposed ERT robot. Externally, it looks like a lizard (with wheels instead of legs). It has an angular head, a body or car, and a tail. The tail keeps the robot in balance during flight. The robot has a retractable propeller on the back of its body or car. There are two extendable effectors on both sides of its body.
The head of the robot contains headlights for path planning and potential vision requirements. The whiskers enable the robot to avoid obstacles and determine if a crack is wide enough for its body to pass through, much like the whiskers of a cat. There is an extra one on the top of the head to test for height. These whiskers would be useful during flight if it can be extended to match the propeller height. There is a heat sensor underneath the head to protect itself from overheating. If the heat sensor indicates a certain temperature threshold been reached, the robot activates the retractable propeller to fly pass the “problem area.” A propeller on each car elevates the car. The back rotor keeps the robot moving forward until the temperature sensor indicates the temperature has fallen below the maximum threshold.

Since flying is draining on the batteries, the robot will descend at a specific time-frame to check if the new area is possible to land, and ascends back if the temperature threshold is reached again. In order to make sure the temperature of the whole land-
ing area is big enough for its body (that is, the head may pick up a low temperature but
the body section is still too hot to land), each car has a heat sensor below it.

Figure 12 shows the head of the robot (a) and a potential flight path of an ERT
robot in relation to height (b). If the temperature at a location is high, the robot keeps
ascending until it reaches a safe temperature threshold. However, the relationship
between (robot) elevation and temperature varies from one location to another. ERT
robots have to ascend to a much higher elevation at a location with fire than a location
without a fire.

The head also contains all other sensors to locate victims such as pulse or tri-
age sensors. Cameras (vision) and sound detectors are all on the head, much like the eyes
and ears of a lizard.

**Body and cargo area.** Figure 13 shows the body (or car) and the internal cargo
system. The first car is connected to the head. The add-on cars are “slave” cars while the
first car is the “master” car, like the hard drives setup in a computer.

The cargo is the medical supplies. The system board, battery, and so forth, are
in the middle of the track. The supplies are kept in containers attached to a conveyor belt.
Each container is predefined to hold a specific medical item. For example, containers one through five may be water packs, while containers six to ten are gas masks. The robot rotates the conveyor belt to the desired item and dispenses it from a portal in the front “chest” or the upper part of its “belly.”

This design minimizes the robot’s need to identify the item through vision or any other complicated method. This would require only a simple inventory-tracking program. The design adheres to the goal of a creating a lightweight and disposable ERT robot.

An alternative to the above design is to use barcode identifications for the items. The robot can scan the barcode to identify the objects. However, the barcode method is not as efficient, as the robot will have to scan a whole batch of items to find the desired one.

**Chassis extension.** The ERT robot chassis can be extended, as shown in Figure 14. This thesis alternatively calls the chassis a *car*, like the car of a train. Each car
is linked to the previous car by the human rescue team, before dispatch. The ERT robot knowledge base is updated with the number of cars it has and the supplies it carries. Each car has an identifier so that the robot can keep track of its inventory.

Tail. The tail is for stability during flight. It has a propeller and two side stabilizers. The stabilizers can be fixed or they can rotate on shafts to change the angles of incident. An internal leveling sensor sends signals to the stabilizer controllers to adjust the stabilizers accordingly.
Power. It is difficult at theory level to access the power requirements for ERT robots. Many aeronautical robots use gas-powered engines [19], [87]. If ERT robots use gas-powered engine, the batteries will be used for operating other features such as cargo door, wheels, conveyor belt, and so forth. The actual battery consumption will depend on the implementer.

Software Design Requirement

The ERT robot is a medical response robot. It is an extension of a mobile robot. Hence, the ERT robot program would need basic path planning, sensor data handling, and a medical decision support program. Discussed are two different aspects in its software design: the software consideration to fulfill the goal of the robot and the issues involved in controlling the robot itself.

Software Considerations in ERT Robot Designs

Three questions were raised in Chapter 3:

• How do the ERT robots find victims?
• How do the ERT robots interact with victims?
• How do ERT robots decide what to give victims?

**How does the ERT robot find victims?** There are many ways to do this in search and rescue robots. A robot can rely on computer vision and image processing, heat sensor, sound, life sign indicator, and so forth. The author proposes using a pulse indicator and sound as a confirmation. The reason is that image processing in a dynamic environment is very tenuous, and may be unreliable. Heat sensors are not reliable if the environment has fire or something hot nearby. One may argue that a temperature range from 97 to 103 degree Fahrenheit is an indicator of a life sign. A pool of water from a broken water cooler could be heated to that level if it is near a fire. In order to make sure it is a life form, image processing would be required. However, as mentioned earlier, image processing may be unreliable. This means ERT robots would need to add more sensors and algorithms for sensor data fusion in order to identify life signs.

A triage or pulse sensor is a good option in locating victims. There are many animals in nature that use non-visual sensors for locating its food. One such animal is the golden mole living in deserts. The mole is blind and it uses vibration (sonar) to maneuver under the hot desert. It finds its food (termites) by feeling the vibration of the grass [62]. Termites are usually found under the grass. Thus, it is possible that if an environment has wireless interference (or other vibrations), pulse signals may still be unique enough for the robot to identify them.

Triage or life-sign sensors might also be cheaper to operate than vision in terms of monetary cost and data processing requirements. It would be an interesting research to compare the two methods.
How does the ERT robot interact with victims? The robot will follow the guidelines for a human ERT member. Victim assessment protocols [1],[2] include:

- bumping the victim with a foot to check for response. In an ERT robot’s case, it can bump the victim with the robot body.
- talking to the victim. In the ERT robot’s case, it either uses Text-To-Speech (TTS) or prerecorded sound files.
- checking for pulse. In the ERT robot’s case, on-board pulse sensor or triage monitor.

How does the ERT robot decide what to give victims? The simple answer is to leave a set of everything to each victim identified. However, an efficient program should try to make the maximum benefit to a victim while conserving supplies. This requires a medical decision support system (DSS).

However, using a DSS raises some questions. What should the robot do if a victim wants something the DSS system did not recommend? Should the robot give the victims whatever they want, or, should the robot make the final decision for them, as the victims may not know what they want? Furthermore, what should the robot do if victims want to hog all of the supplies to themselves?

These questions pertain to medical ethics. Therefore, this thesis will not delve into these questions but let the DSS designers decide.

AI in Robot Control Programs

In order for an ERT robot to protect itself from heat, prevent itself from being stuck in a small opening, find its way around in a foreign environment, and interact with victims, it needs to have a certain level of “intelligence”. Various artificial intelligence
techniques have been used in SR robots for navigation, image processing, sensor data fusion (see next section), and decision support system. These techniques are suitable for a very dynamic search and rescue environment.

**Sensor data fusion.** Data from sensors can be overwhelming for the system [42]. Furthermore, considering the amount of sensors involved, there needs to be an efficient way to make sense of all the data that the sensors transmit. Sensor data fusion is a technique that creates a model based on the raw data from various sensors.

In the paper “Principles and Techniques for Sensor Fusion,” Crowley and Demazeau [43, p. 3] wrote:

Perception is not a goal in itself, but a means to obtain a certain behaviour by an agent (a thing which “acts”). In order to plan and execute actions, an intelligent agent must reason about its environment. For this, the agent must have a description of the environment. This description is provided by fusing “perceptions” from different sensing organs (or different interpretation procedures) obtained at different times.

We define *perception* as:

The process of maintaining of an internal description of the external environment. The external environment is that part of the universe which is accessible to the sensors of an agent at an instant in time. In theory, it would seem possible to use the environment itself as the internal model. In practice, this requires an extremely complete and rapid sensing ability. It is far easier to build up a local description from a set of partial sources of information and to exploit the relative continuity of the universe with time in order to combine individual observations.

We refer to the problem of maintaining an internal description of the environment as that of “Dynamic World Modeling.”

Recent research has shown that the extended Bayesian reasoning – Dempster-Shafer theory of evidence is either equally reliable or has surpassed traditional Bayesian techniques [39], [40], [41] in terms of sensor fusion.

Different sensors give different types of data. The data from these sensors are “raw” data. The system cannot operate on raw data for two reasons: One, the amount of
coding required to handle different data is an inefficient use of system resources. Two, when improved sensors become available the system would need to be reprogrammed.

An adapting or normalizing layer is required. This layer would translate or normalize raw data from a sensor into a system acceptable format. This adaptive or normalizing layer must be decoupled from the main system design. The ERT Robot system should normalize the data first and then if necessary, use techniques such as Bayesian network for fusion.

**Software Consideration for the ERT Robot**

This section redirects the focus back to the mobile robot aspect of an ERT robot and discusses two major issues in mobile robot programming: embedded system programming and hardware control programming.

**Working with embedded systems.** Mobile robots run on various platforms and computer boards. Chapter 3 shows robots of various sizes and Chapter 2 shows that most of the mobile robots used in search and rescue are of the smallest sizes. Murphy has pointed out that only shoebox sized SR robots can enter crevices [16]. This size refers to a fully assembled robot. Most desktop computer motherboards would be too big for a system board (some of the more recent laptops might fit in this requirement). This is the reason why many small robots use smaller computer boards.

Embedded system is a term that refers to very small computers in small devices. Most common everyday embedded systems are found in Personal Digital Assistant (PDA), cellular phones, and many other consumer devices or household appliances. Inside the embedded system is a micro-controller board [55].
A micro-controller is a small microprocessor chip that can handle its own I/O and memory management functions [56]. This makes the micro-controller powerful. However, these functions are limited. A regular computer uses a microprocessor and has other supporting chips to handle I/O and memory. Microprocessors allow large amount of memory to be swapped in and out. On the other hand, a programmer is limited to the memory and processing power embedded on micro-controller chips.

Limited memory and processing power are the reasons why many of the techniques used in desktop computer programs would not work on micro-controllers. Such techniques include:

- Keeping a large volume of data in cache to speed up processing. This technique is frequently used to minimize pulling data from database (I/O operation), which is generally slower.

- Relying on databases. There is no hard-drive to store a “database” in micro-controllers as there are in traditional computers. Micro-controllers work with EEPROM chips. An EEPROM chip is an integrated circuit in which both the program and data are “burned” into it. EEPROM chips usually have limited space and read/write limitation.

- Relying on high-level data types for storage or convenience. Embedded system programming language may not support complex data types.

Furthermore, lightweight programming languages or operating systems are required to control robots. Since all the programs and data are contained in the EEPROM chip, programming languages must be written specifically for embedded systems in mind. Many programming languages come with large libraries. Their embedded counter-
parts generally drop these libraries or have them rewritten. Embedded Java from Sun Microsystems and Real-Time Specification for Java (RTSJ) [88] are such examples.

There are other embedded system programming considerations. Optimization techniques may have to be deployed, as embedded programs may have to work with limited memory and a slower processor speed. Embedded programs must be small and efficient.

Another issue to consider is that the robot sensor data may be different from computer program data [41], [43]. Other required techniques may include normalization, typecasting and data formatting.

Controlling hardware. The software program for robots generally looks different from a desktop application. A robot program has to control the robot’s hardware and sensors. For example, it has to send specific electric voltages to devices and rotate motors. It has to calculate the number of revolutions required to make a turn. It has to decide which wheel to move [76].

In a personal computer, hardware peripherals such as printers, disks, and monitors are generally handled between the motherboard, the system BIOS, and the operating systems. Many of these controls are abstracted in a high-level program [77]. If micro-controllers are used in a robot, the programmers must deal with these issues themselves. However, this is beyond writing a low-level layer program like those operating systems, but that hardware control is actually an integral part of a micro-controller based robot system. This is a paradigm shift for robot programmers and must be considered carefully during system design. Some examples include:
• Battery consumption—what (power draining) sensors and devices should be kept on and when should they be shut off?

• Motion—what are the numbers and angles of rotation (robot wheels) required to move the robot to a target location?

• Image processing—what is the angle of rotation for a camera required to identify a target? Furthermore, when should the camera adjust its focus?

The list of considerations for ERT robots grows as the control of cargo doors, the dispatchment of its cargo, the control of its propellers, and the control of its rotors when it decides to fly, are added.
CHAPTER V

MODULE DESIGN AND IMPLEMENTATION

This chapter proposes an ERT robot system design and implementation. The chapter is broken down into four sub-topics:

• Designing an architecture for an ERT robot system
• Using a non-visual method to locate victims
• Demonstrating a sample interactive and diagnostic program
• Communicating with other robots

System Architecture and Design

Figure 15 presents the ERT robot architecture in abstract layers and Figure 16 shows the types of modules in each layer and their relationships under ideal conditions.

In Figure 15, the first or highest layer is the sensor layer. This layer represents the actual sensors equipped on the robot. The second layer is a normalizing or adaptive layer. As mentioned above, sensor data comes in different formats. In order to use the data effectively, the various formats should be normalized to a single format. The normalizer must be carefully constructed so that it does not lose the original data’s integrity, as normalizing different data formats introduces a risk of losing vital data in exchange for interoperability [78]. The third layer is an analysis layer. It processes and analyzes the data to decide what action to take. Once it decides upon an action, it triggers the last layer, the Action layer, to implement the action.
Fig. 15. Layer architecture.
The Action layer can also be considered a low-level layer, as the robot action will most probably be involved in doing something physical, such as moving or grabbing objects. After an action is performed, the system loops back to gather more data from the Sensor layer unless the system has completed or aborted its program.

Figure 16 presents a breakdown of the layers to show the different modules involved. Notice that the sensors are broken into two tracks – Victim Identification and Assessment module and Self-preservation module at the Analysis layer. Hence, the robot is really doing two things, first, locating and dealing with victims, and second, finding its way around while protecting itself.

Modules Overview

There are five main modules in the system: Vital Sign Analysis, Victim Interaction, Action, Search and Navigate, and Self-preservation.

**Vital Sign Analysis module.** This module reads the data from the camera, microphones, pulse, or triage sensors. It then processes this data to infer if a victim is located. Once the inference engine concludes that a victim may have been found, the VictimIdentified state is updated. The Vital Signal Analysis module then passes the control to the Victim Interaction Protocol module.

Many sensors have a certain percentage of errors in their data [79]. When these sensors are used in an environment where signal interference exists, the module would not be able to get accurate data. This thesis suggests using fuzzy logic techniques as a solution to this problem. If the fuzzy inference module infers that a given set of data is not describing a “victim,” the robot resets all victim states to default and sets a “ready to move on” indicator or flag to the master state machine.
Victim Interaction module. The Victim interaction module incorporates the standard ERT protocols [1], [2] such as bumping disaster victims or talking to them. The Victim State machine is updated after each step in the protocol. After giving victims their appropriate supplies, the robot will move on to locate the next victim.

In order to communicate with victims, the ERT Robot would need some kind of communication capability. The ERT robots need Natural Language Processing (NLP) [12] capabilities in order to communicate with victims via human language. (See the following section for a prototype Victim Interaction module.)

![ERT decision modules diagram](image_url)

Fig 16. ERT decision modules.

Action module. This module contains many sub-modules or controllers for navigation, image processing, and controllers for effectors and actuators. This module...
provides low-level control of the robot. The Self-preservation and Victim Interaction Protocol modules make action requests to this module.

The Action module is also in charge of navigation and motion. Therefore, once the Victim State machine moves to the “Finish with Victim” state (Figure 17), the module moves robot on to search for the next victim.

Search and Navigate module. The Search and Navigate module enables the robot to find its way in an environment. This module requires path planning, obstacle avoidance, and target analysis algorithms. Since the focus of this thesis is not on these topics, it will not discuss them.

Self-preservation module. The main function of this module is to protect itself from damage, but that does not include defending itself from being attacked by victims.

One example of self-preservation, as defined in this thesis, is when the heat sensors (under the robot head and cars) indicate that the temperature has reached a certain threshold. The Self-preservation module immediately sends a signal to the Action module to elevate the robot. The Self-preservation module can be expanded and equipped with various sensors to protect the robot.

This logic in the Self-preservation module mimics human and animal reflexes. As illustrated above, if the heat sensors reach a certain threshold, the module will immediately send an elevation request to the Action module to elevate the robot. Unlike the Vital Sign Analysis module, this module does not require fuzzy logic inference engine. IF-THEN-ELSE statements should be sufficient for this module.
State Transitions

The Victim Interaction, Search and Navigate, and Action modules have internal and external states that record information about the robot and its environment. The Master State combines local state information from the Victim Location and Assessment module and the Self-preservation module. Local states are reset when their cycle has completed. Discussed below are two sample state diagrams for the victim assessment track and the robot functionality track.

Victim State diagram. Figure 17 is a state representation of the Victim State Machine. The state machine begins at the Start state. Then, it gathers sensor data at a specific interval of time. The data is normalized and forwarded to the IdentifyVictim state. The IdentifyVictim state will produce a true or false value only.
A True transition moves the state machine to the InteractWithVictim state. The system can stay in this state indefinitely, and will only transit out when a VictimHelped event is triggered.

A False transition from the IdentifyState moves the state machine to the Reset state. This final state will reset the victim state machine. This completes the victim location and interaction cycle and the state machine will loop back to the Start state until the robot’s mission is completed.
Robot State diagram. Figure 18 is a state representation of the Robot State machine. Like Figure 17, the state machine begins at a Start state and moves through the Normalization state. Next, the state machine moves to Determine Action state. As mentioned in Chapter 4, the Self-preservation module mimics animals’ reflex responses. Thus, each sensor is tied to a reflex action.

Fig. 18. Robot functionality.

Once an action is determined, the PerformAction state processes the action in the Action module. The system can stay in this state indefinitely. The only transition out is the ActionFailed or ActionComplete event. If the action fails, the system loops back to the start state to gather new sensor data and check for environmental
changes. On the other hand, if the action completes, the state machine is reset and the
Self-preservation state machine is back at the Start state.

If the DetermineAction state outcome is NoActionNeeded, then the system loops back to the Start state.

Locating Victims Using a Non-visual Method

This section presents a hardware prototype that demonstrates how to locate a victim without using computer vision (camera). The purpose of this demonstration is that image processing is expensive and problematic [30]. Furthermore, CRASAR research has demonstrated that victims can be located using non-visual methods [16]. They use a triage sensor that picks up biological signals such as pulse and breathing.

One may question how the victims are located without vision in the first place. It is assumed that the victim is already located by another robot and an identification tag has been left behind. The identification tag used in this demonstration is a simple IR beacon. (Note that there are other wireless techniques such as radio transmitters that can also be used as beacons. Finding a non-visual method to locate victims is a research area.) Chapter 3 argued that SR robots should be organized as a role-based robot team and emphasized that ERT robots are meant for rescue, and not for search. In a collaborative effort, the scout robots will leave the identification tags for the rescue robots. In this way, the hard work of locating victims through path planning and obstacle avoidance will only need to be done once per victim.

In the prototype demonstration, a “victim beacon” emits infrared signals and the prototype ERT robot will detect and follow these signals to locate the “victim.”
Prototype Materials

The materials used in the prototype robot are arbitrary. Figure 19 shows the robot with and without headlights. The on-board headlights enable the demonstration to be held in a dark room. The IR emitter and the tether wire in Figure 20 (center) is the “victim beacon.”

Fig. 19. ERT robot prototype.
The robot prototype is built from the Parallax BoeBot kit [29]. It contains a Basic Stamp II micro-controller, a Board of Education System board, a serial cable, and a few integrated circuits and components. For demonstration purposes, the IR emitter and the robot share a single system board.

The robot control is programmed in Parallax Basic programming language [29]. One section of the code sends out IR signals every 300 milliseconds. The other section of the code listens for the signals and controls the robot movements. During the prototype testing, the infrared emitter is placed five feet away from the robot. The on-board IR detectors detect the signals and notify the robot program. The program then moves the robot towards the signals.
Robot Cost

The prototype costs $160 to $240. The reason for the price range is that an optional wireless camera can be used to record the prototype robot in action. The following is a breakdown of the cost:

- BoeBot - $150 (from BoeBot kit).
- Headlight casings - $3 (extracted from 2 key chain flashlights)
- Headlight bulbs - $2.79 each
- IR detector - Comes with BoeBot kit
- IR transmitter - Comes with BoeBot kit
- Wireless Camera (optional) - $80

Issues with Implementing the Proposed Hardware Design

There are many features outlined in the ERT robot hardware design. The biggest problem is acquiring parts. One option is to build a part from the ground up. The other option is to get existing parts from toys, miniature models or replicas, machines, or electronics.

Even though the hardware cost adds up very quickly, the ERT robot prototype presented in this chapter can perform some basic functions under a $300 retail price. The most expensive feature for ERT robots is flight. The components in many radio-controlled helicopters are not suitable. The rotors and engines are not strong enough to lift a few pounds. USC acquired a $1700 gas-engine helicopter for their UAV robot. However, they still had to upgrade the engine to a stronger one [69]. There needs to be a cheaper way to enable robots to fly. This is an area worth investigating.
The thesis proposed many requirements for an ERT robot. Table 6 lists the problems the implementers would face if they construct an ERT robot from the listed components.

Sample Interactive and Diagnostic Program

In order to help the human victims, ERT robots have to communicate with them, diagnose their physical conditions, identify their physical needs, and decide the appropriate medical item to give them. The purpose of the Victim Interaction module is to fulfill these functions. This subsection addresses the issues involved in designing the module and presents a sample implementation of it.

Communicating with Humans

Interacting with human victims requires natural language. However, adding language-processing capability to an ERT robot system has some problems. First, there are some issues with using NLP. Processing a user input requires syntactic processing and semantic analysis [12]. The Earley parser (a type of dynamic algorithm) used in NLP produces tree outputs. After the trees are generated, they must go through semantic processing. There are two approaches to semantic processing: machine learning or dictionary lookup [12]. If the ERT systems use machine learning, they would require a large memory cache, as machine learning requires a different classifier for each input variable. These classifiers get large very quickly as the number of input variable increases.

\[
\begin{array}{|c|c|}
\hline
\text{Feature} & \text{Enabling the robot to fly} \\
\hline
\end{array}
\]

TABLE 6

ISSUES WITH CONSTRUCTING A LOW COST ERT ROBOT
**Feature**  | **Heat sensors**  
--- | ---  
**Attempt**  | A basic thermistor from Radio shack.  
**Problem**  | The thermistor did not respond to heat.  
**Conclusion**  | This is an easy oversight, as there are many simple IC components such as resistors and capacitors that do their function well. A robot actually requires a heat module. A thermistor is not enough.  

**Feature**  | A non-image method of identifying a victim (life form) from an object.  
--- | ---  
**Attempt**  | Searched the internet for a non-invasive pulse sensor.  
**Problem**  | The closest is the University of Sussex in Brighton research that is about to launch the product [70]. Other option is the triage sensor from CRASAR. The price is unknown.  
**Conclusion**  | Creating a low cost non-invasive pulse sensor is a research area.  

**Feature**  | Speech recognition and synthesis devices.  
--- | ---  
**Attempt**  | Two low cost voice recognition module from Sensory Inc - Voice Direct and Voice Extreme.  
**Problem**  | Voice Direct is voice dependent. The sibling - Voice extreme is not and has a proprietary C library.  
**Conclusion**  | Voice Extreme can handle up to 2Mb of code, so the user interaction software module could be included.  

**Feature**  | A camera to mount on robot to see what the robot is doing.  
--- | ---  
**Attempt**  | A wireless camera or webcam to view the robot’s movement.  
**Problem**  | Different wireless cameras have various image resolution quality.  
**Conclusion**  | Many cameras can save images in bitmap or some form of compression format such as JPEG or GIF. The CMU cam is a popular choice for robotic image processing and it costs around $150.  

**Formal Attempt**  | A $50 battery-operated Raptor with Styrofoam chassis and plastic rotors.  
**Problem**  | The RC helicopter weighs about five ounces and can only carry at most a few ounces of payload. The plastic (pvc) rotors and the motors are too weak. Even if the motors are upgraded, the plastic rotors are too weak. USC modified the Kyosho Concept 60 RC helicopter and upgraded its engine to Enya 80. The modified helicopter can carry a payload of 9 pounds [69]. The Kyosho Concept 60 rotors are made of composite material and the Enya motor is a gas motor. The retail price for Kyosho Concept 60 is $1700, the motor cost about $180.  
**Conclusion**  | The Concept 60 is too expensive for a disposable robot.
dictionary approach is less computationally intensive as it only requires the program to look up a word [11]. However, it requires a database. If an embedded system is used for the robot, adding a database is not possible.

Another problem with adding language capability is the need for an association between the victim’s input and a system task. These associations can be hard-coded. Alternatively, an Expert System or Case-Based Reasoning (CBR) [34], [35] model can be used. The ERT robot agent then processes the request based on past or trained cases.

The problems with NLP become even more complicated when confused victims ramble, or use vague or broad terms like *bad, good, nearby*. Even though vague or broad terms are not part of traditional data, they are still data. They are meta-data, a special type of data that describes other data. However, adding these types of data into the system creates two problems: first, it is very tedious for a human to input all possible meta-data or equivalent terms for a word into the database, not to mention some terms will be overlooked. Second, entering every possible equivalent term will create a very large database. This increases the response time for a database’s query and creates large result sets. The ERT robot agent system will be bogged down with computation. This affects the agent’s reasoning ability and efficiency. The ERT robot can use fuzzy logic techniques at the language’s word level to handle these issues [64].

The ERT robot has to maneuver in a potentially hostile and foreign environment that is not programmed into its *world* state. It has to rely on whatever information it can gather in real time and make an inference on its environment. At a disaster site, everything may be mish-mashed together (i.e., body parts covered in dust with protruding wires) [37]. Fuzzy (and probabilistic) reasoning is very useful in this situation.
Problems involved in communicating with victims. As presented in the previous section, NLP-based communication module is quite computationally intensive. Thus, an ERT robot should not maintain a lengthy conversation with victims. One method to resolve this is to cut down the robot’s side of the conversation by having pre-recorded sound files. The robot can then focus on Automatic Speech Recognition (ASR). When victims say something that the system is waiting for, the robot can play an appropriate sound file.

There are also other communication issues. Environmental noises can confuse the robot as to what the victim actually said. This can lead the robot to make an inaccurate decision.

Sample Algorithm and Reasoning Model

This section presents a sample algorithm and reasoning for an ERT medical decision support system. Figure 21 shows an algorithm suitable for dispatching medical items.

The algorithm is called after a victim is identified and the victim assessment protocol [1], [2] has been performed. The only value passed in is whether the victim made any physical movements. The algorithm requires this value because the outcome of it is to identify the medical supplies to release.

Figure 22 shows a sample reasoning model that dispatches medical supplies to victims. The model uses observed symptoms or behaviors to infer the appropriate
algorithm IdentifySupplyDistribution(boolean bodyResponse)
    Array condition;
    List defaultItem, item;

    defaultItem = recommendMedicalSupply();
    IF bodyResponse is true THEN        verbalResponse = talk()
        IF verbalResponse is true THEN          condition = askVictimCondition()
            item = recommendMedicalSupply(condition);
        ELSE
            Add RFID_Tag to item

        FOR i=1 to defaultItem.length
            if defaultItem[i] is NOT in item then
                Add defaultItem[i] to item
        LOOP
    END IF
    return item

Fig 21. A sample algorithm for supply distribution.

medical items to give to victims. Then the model calls a sub-function to locate the item in
the cargo cars. If two or more cars have that item, the first location is used and the state
machine is updated.

If the victim is alive but not moving, a victim identification tag can be left
behind. Note that the algorithm includes an optional line that dispatches a default item for
every victim. The algorithm does not loop because the system control should be at a
higher level.

Sample Module Implementation

Figure 23 shows a mock interaction between an ERT robot prototype system
and a “victim.” For simplicity, the demonstration is performed on a Personal Computer
(PC). The prototype does not include any speech synthesis or recognition. It focuses on
**Step 1:** Gather Data at time interval (sample data below)

- Symptom or Behavior 1 (70% confident or probable)
- Symptom or Behavior 2 (90% confident or probable)
- Symptom or Behavior 3 (50% confident or probable)
- Symptom or Behavior 4 (50% confident or probable)

**Step 2:**

Pull out from knowledge base all cases where symptom/behavior 2 exists (maximum confidence).
Assign all these cases (num_of_cases=set A) to confidence level 0.5.
IF (num_of_cases > acceptable_num) THEN
    refine result to include symptom/behavior 2.
    Assign the narrowed cases (set B) to confidence level
    \[0.9 \times 0.7 = 0.63\]

IF (num_of_cases(setB) > acceptable_num) THEN
    refine results to include symptom/behavior 3 and 4
    Assign the narrowed cases (set C), where
    confidence level = \(0.9 \times 0.7 \times 0.5 \times 0.5 = 0.15\)

    cases_monitored = setC
ELSE
    cases_monitored = setB
ELSE
    cases_monitored = setA
ENDIF
IF car is not empty THEN
    select (condition)
    case: setA
        item = itemA;
    case: setB
        item = itemB;
    case: setC
        item = itemC;
    car = locateCarForSupply(item);
    expulse(car, item);
ENDIF

**Step 3:**

Update state machines

---

Figure 22. A sample reasoning model to identify and locate supplies.
the user interaction logic instead. The User Interaction module prototype uses the Knowledge Worker (KW) agent model [64]. The open source project called Java Personal Agent (JPA) at SourceForge.net [80] is built on the KW agent model. JPA is written in Java, but can be written in another language, such as VE-C in Voice Extreme. The JPA program has its own natural language processing engine. The engine uses the Earley algorithm to parse user input. Figure 24 shows the JPA model.

There are two classes in the software prototype. The UserInteractionMgr class demonstrates an interface that is in charge of interacting with, accessing conditions, and
deciding treatments (as above, the treatment recommendations may come from another application or class) for a victim. Note that the actual triage assessments and treatment recommendations are not performed by this class.

The CargoManager class represents a controller that handles low-level hardware control, such as opening a specific cargo door. It also updates an internal inventory state to notify the UserInteractionMgr when an inventory runs out.

When a new KW manager is added to the system, the caseLookup.xml file must be updated (the caseLookup.xml file is part of the KW agent system). The next section shows an XML code snippet for the UserInteractionMgr module. This section also shows a configuration file called CaseLookup.xml.

**User interaction manager class.** Since the prototype uses the KW agent model, the UserInteractionMgr class is a KW Agent manager. All KW Agent managers must extend the GenericManager class [80]. The following is the code from the UserInteractionMgr.java file.
public class UserInteractionMgr extends GenericManager{

    static LogFile lf = new LogFile().getLogHandle();

    public Object[] findBySpecifiedSet(Object[] set, 
                                      Constraint[] fc, 
                                      int compareType) throws Exception{
        String msg = "This is currently not part of my 
                     functions";
        return reply(msg);
    }

    public CBResult[] accessNeed(Constraint[] fc, int 
                                  compareType){
        String msg = "My name is PA. I am an Emergency Response 
                     Robot. I 
                     have water, bandage and some food on me. What do you 
                     need?";
        return reply(msg);
    }

    public CBResult[] processUserRequest(Constraint[] fc, 
                                          int compareType){

        String msg = "";
        String item = null; //default
        String itemDispatched ="";

        for (int i=0;i<fc.length;i++){
            String cond = fc[i].getName();
            if (cond.equalsIgnoreCase("thirst"))
                item="water";
            else if (cond.equalsIgnoreCase("bleed"))
                item="bandage";
            else if (cond.equalsIgnoreCase("breathe"))
                item="mask";
            else if (cond.equalsIgnoreCase("food"))
                item="health_bar";

            if ((item!=null) && (!itemDispatched.equals(item))){
                lf.log("User requested item =" + item);
                itemDispatched = item;
                msg = CargoManager.dispatchCargo(itemDispatched) + " ";
            }
        }
        lf.log(msg);
    }
}
Only two methods are part of the UserInteractionMgr class. They are the accessNeed and processUserRequest methods. The remaining methods are mandatory for any class that extends the GenericManager class.

- **accessNeed()**—The method identifies what the user wants or needs. In the demonstration code, the victim is assumed to be coherent and the agent merely talks to the victim to identify their need. In practice, if the victim is unconscious or confused, this method would call other helper classes to perform triage. In practice, this method should only assess the condition of the victim and not the treatment.

- **processUserRequest()**—This method contains the “medical treatment” reasoning logic. The method infers an item to dispatch based on the victim condition. In the prototype, the victim’s condition is fairly straightforward and she is able to verbalize her problem. The processUserRequest method infers an appropriate item for her and calls the CargoManager class to “dispatch” those items.

5.3.3.2 *Cargo manager class.* The following is the code from the CargoManager.java file. It shows how the manager decides which cargo to dispatch.
public class CargoManager {

    static LogFile lf = new LogFile().getLogHandle();
    static boolean initialized = false;

    //this is the master inventory list
    static Hashtable allInventory = new Hashtable();

    //this method to initialise inventory.
    //Here we hard-code for demonstration purpose
    static void init(){

        //the inventory is categorized from item names as it is
        //what user want. We don’t care which car has it.
        //Alternative to the LinkedList class. LinkedList
        //has no special advantage over Vector.

        lf.log("loading cargo into cars...");

        Vector water = new Vector();
        water.add("car_1");
        water.add("car_2");
        water.add("car_3");

        Vector bandage = new Vector();
        bandage.add("car_1");
        bandage.add("car_3");

        Vector mask = new Vector();
        mask.add("car_2");
        mask.add("car_3");

        Vector food = new Vector();
        food.add("car_2");

        allInventory.put("water",water);
        allInventory.put("bandage",bandage);
        allInventory.put("mask",mask);
        allInventory.put("health_bar",food);

        initialized = true;
        lf.log("done");
    }

    //This method simulates how to dispatch cargo
    public static String dispatchCargo(String item){

    }
String result;

if (initialized==false)
    init();

lf.log("processing cargo dispatching request");

String car = locateCargo(item);
lf.log("car " + car);
if (car==null)
    result = "No more " + item + " left";
else
    result = releaseCargo(car,item);

lf.log(result);
return result;
}

//this method searches all available cars and returns the first car
//that has the item. The master inventory is updated.
static String locateCargo(String item){

Vector cars;
String car = null;

//find first car that has the item here
//if can’t find it, set car =null

lf.log("locating item " + item + " in cars...");
Object obj = allInventory.get(item);
if (obj !=null){
cars = (Vector)obj;
if (cars.size()>0){
    Object cr = cars.get(0); //get the first object
    car = (String)cr;
}
}
return car;
}

//this method is a simulation of actual hardware control
to release // the object from a car

static String releaseCargo(String car,String item){
/*
  * some hardware control happening here
  */

//get hold of the item list
Object obj = allInventory.get(item);
lf.log("inventory obj " + obj);
Vector itemList = (Vector)obj;

//remove a car’s reference once the item is dispatched from it
itemList.remove(car);
String msg = "Cargo " + item + " released.";
return msg;
}

The class has three main methods. They are dispatchCargo(), locateCargo() and releaseCargo().

- dispatchCargo—This public method interfaces with the UserInteractionMgr class to release an inventory item for the user.

- locateCargo—This internal method searches all the ERT robot cargo cars to locate the item. An item may exist in more than one car. The first car found with the item is used. If the requested item does not exist or has ran out, the method returns the message to the dispatchCargo method.

- releaseCargo—This method simulates a low level hardware controller class. After the method “dispatches” the item requested, it updates the inventory state.

The init() method simulates a cargo inventory loader. For demonstration purposes, the inventories are hard-coded into the inventory state.

CaseLookup XML file. All KW agent managers must configure the CaseLookup.xml file in order to be part of the model. The software prototype manager
for an ERT robot is represented by the UserInteractionMgr class. The following is a sample XML code snippet for the UserInteractionMgr class.

```xml
<case>
  <keywords>
    <keyword>user;me;help</keyword>
    <keyword>help</keyword>
  </keywords>
  <constraints/>
  <manager name="pa.cbr.cases.UserInteractionMgr">
    <method name="accessNeed">
      <argType>Constraint</argType>
      <argType>int</argType>
    </method>
  </manager>
</case>
<case>
  <keywords>
    <keyword>user</keyword>
    <keyword>thirsty;water;air;breathe;bleeding;hurt;hungry;food</keyword>
  </keywords>
  <constraints>
    <target name="thirsty;water">
      <constraintType name="text">
        <constraintName>thirst</constraintName>
      </constraintType>
    </target>
    <target name="air;breathe">
      <constraintType name="text">
        <constraintName>breathe</constraintName>
      </constraintType>
    </target>
    <target name="bleeding;hurt">
      <constraintType name="text">
        <constraintName>bleed</constraintName>
      </constraintType>
    </target>
    <target name="hungry;food">
      <constraintType name="text">
        <constraintName>food</constraintName>
      </constraintType>
    </target>
  </constraints>
  <manager name="pa.cbr.cases.UserInteractionMgr">
    <method name="processUserRequest">
```
Collaboration with Other Robots

Disaster sites usually have many USR robots. The rescue efforts can be greatly improved if the robots could work together. In Chapter 3, it argued that USR robots should be a role-based group effort. Designing a system that enables collaboration between USR or ERT robots could increase the odds of finding and rescuing victims. These kinds of collaboration systems require distributed or multi-agent system designs. There are various definitions on what is considered multi-agent systems. Multi-agents systems can be considered as a type of distributed system [44], [45]. Multi-agents system is a challenging field. Many researches have been done on this topic and various techniques have been introduced. Thus, there exists many multi-agents system architectures [48] and communication techniques for distributed robotic systems [46], [47]. These are beyond the scope of this thesis.

Communication Design Issues

This thesis does not propose a multi-agent communication system design for ERT robots. There are two reasons for this. One, since the thesis proposes splitting the search and rescue robots into roles, there will be different types of SR robots from different manufacturers or research facilities at the same site. As mentioned above and in Chapter 3, the current USR robots have different systems and protocols. Without a common communication protocol, they would not be able to communicate. Two, collapsed
buildings at disaster sites usually have a lot of wireless interferences from debris such as concrete blocks. They create problems in wireless communication. Unless this problem is addressed, whether by adding signal relays or noise management techniques, communication between the same type of robots (for example, between two ERT robots) may not even be possible.
CHAPTER VI

CONCLUSION

Review of Discussion

This is a recap of what was discussed in this thesis. The author discussed in:

• Chapter 1 the world of search and rescue robots along with the need for a special search and rescue robot.

• Chapter 2 the background information required to understand the issues involved in Search and Rescue (SR) robots.

• Chapter 3 some of the current search and rescue robots. It also argued that researchers should shift the paradigm on what constitutes as a search and rescue robot team.

• Chapter 4 about payload, presented the hardware and software design for an Emergency Response Team (ERT) robot, and addressed the questions raised in the previous chapters.

• Chapter 5 the layer and module architectures for an ERT robot system. It then presented the prototypes built using the suggested design.

Main Points for the Design of an Emergency Response Team Robot

To reiterate the main points of the proposed ERT robot:

• In a role-based SR team, its role is a medic robot. Furthermore, its main function is to deliver medical and emergency supplies to a victim.
• It must be small and lightweight.

• It must be able to access small spaces.

• It must be "disposable." That is, it should cost very little to build.

• It does not and should not have too many sensors. Furthermore, it does not need complicated vision, path planning, and sensor fusion techniques. If there are any, it should be kept at a minimum.

• It identifies a victim based on life signs, rather than image processing.

• It is designed to be untethered and autonomous. However, it can have tether and operator override options.

• Its Human Computer Interface should be user-friendly.

Research Recommendation

This thesis touched on many different issues in search and rescue robotics and ERT robots. There are a few major issues that require additional research, even though some issues are already being actively researched elsewhere [4], [7], [9], [10], [13], [14], [15], [26], [38], [45], [46], [47], [50], [66], [89].

First, enabling multi-agent communication between different USR robots is crucial to a collaborative robot team. A standardized communication protocol is most likely required.
Second, research on role-based SR robotic teams should be reviewed. A role-based SR robot team not only mimics real-life search and rescue teams, but it also follows the key concepts in multi-agent and distributed systems. If the groups involved in USR robot designs work together and focus efforts on a specific role, they would make the search and rescue efforts more efficient; which in turn benefit the victims.

Third, finding a way to incorporate both gas engines and batteries for ERT robots could help extend battery life and rescue duration. Electric motors that control locomotion and flight could be replaced by gas-powered engines. The on-board battery could then be reserved for running other components and devices. If ERT robots use the same principles as cars, where battery (electricity) is mainly powered by engine motion; the on-board batteries could last much longer. Furthermore, an investigation of hybrid engines (like hybrid cars) for ERT robots is also worth pursuing. Many of the problems faced in automobiles and airplanes may converge with ERT robot issues.

Fourth, in Chapter 5 it was noted that ERT robots can have more than one car (module). A single-car (mode) ERT robot looks like a helicopter robot with rectangular chassis. Perhaps only aviation experts can answer whether enabling ERT robots to fly is a difficult problem; especially when robots with multiple cars \(n\) modules flying is not seen in current literature. There may be unforeseen problems with this configuration.

Fifth, interacting with victims is a very important feature for ERT robots. However, NLP is computationally intensive. This is a very challenging problem for microprocessors. To complicate matters, confused victims may also garble and disaster sites may also be noisy.
Conclusion

Research in SR robots is a very challenging and rewarding field. However, it is still a very young field. Research has advanced SR robotics technology over the past few years, particularly from the feedback from CRASAR. It is tempting to build a super SR robot. However, Murphy's experience from WTC shows that bigger (complicated) is not necessarily better [16], [17], [49]. Unlike research robots, SR robots should be designed to be part of real-life rescue missions. SR robots are still tools, albeit intelligent ones. Keeping that focus will enable us to design useful robots that will benefit society.

Furthermore, if one day the idea of ERT robots could be realized and implemented, the questions of whom should control ERT robots and where should they be located will come to play. Should ERT robots reside at customer sites and come alive when an emergency arises (like the Mostitech security robots [71]) or should they be dispatched and controlled by an organization like CRASAR? These are just some of the important questions that need to be addressed as ERT robots become available to the public.
REFERENCES


